



# **Methodology and Tools for Realising Product Service Systems for Consumer Products**

Xiaoyu Yang

Submitted in partial fulfilment of the requirements for the award of

**Doctor of Philosophy**

June, 2006

Faculty of Computing Science and Engineering

De Montfort University



## Declaration

I hereby declare that no part of material described in the thesis has been submitted for award of any other degree or diploma in this or any other university or college of advanced education



## Acknowledgements

There are many people without whom this work would not have been possible.

First of all, I would like to express my sincere gratitude to my supervisors Professor Philip Moore, Dr. Junsheng Pu and Mr. C.B. Wong for their tutorship and valuable support. I also like to express my gratitude to Dr. Harish C. Sharma for his great help with the thesis as well as constructive suggestions.

I would like to thank European Commission for funding the ELIMA project through which I had an opportunity to work at De Montfort University and conducted my PhD research study. Many thanks also go to ELIMA consortium, which includes SONY International Europe, Motorola, Indesit Company UK, Cybernetix, SAT, ReUse, Sheffield Hallem University and De Montfort University.

My special thanks go to my wife, Dr. Sharon X. Lin, who gives me great support and takes a lot of responsibilities for our family in London, so that I can concentrate on my work and PhD research study at De Montfort University in Leicester.

I wish to thank my colleagues Dr. S.K. Chong, Dr. X. Chen and project industrial partner Dr. K. Bodenhofer from SONY International Europe for their collaboration and help in ELIMA project as well as my PhD research study. In addition, I also like to thank each member of the ELIMA consortium and those who helped me to complete the research work.

Finally I would like to show my great appreciation to my parents who always give me great support, and my little baby, Tiffany, who brings a lot of pleasure to my family.



## Synopsis

In response to the drive for the sustainable development which advocates reduction of resource consumption while delivering better and more widely available goods and services, developing Product Service Systems (PSS) becomes an increasingly important strategy. PSS are most commonly defined as “a marketable set of products and services capable of jointly fulfilling a consumer need” (Goedkoop et al, 1999). PSS can facilitate the realisation of sustainability through provision of product service mix and advocating a new business model – “to have a utility instead of an ownership”. In this thesis, an engineering methodology for realising PSS for consumer products is proposed, and associated tools have been formulated to facilitate the realisation.

Currently there are some limited examples of engineering methods and tools which seek the design, development, and production of services and aim at exploring the development of PSS. However, they are unlikely to help industrial manufacturers practically implement and realise sustainable PSS solutions as they are mainly concerned with the general design and development of service and PSS. In the domain of consumer goods, there is a lack of awareness about successful PSS solutions. In order to address these issues, an investigation of engineering methods and tools for the realisation of PSS for consumer products has been carried out in association with the European Commissions Framework V funded project entitled Environmental Life-cycle Information Management and Acquisition for consumer products (ELIMA).

A service enabling scheme which actively utilises product life cycle data in a systematic and integrated manner to facilitate the creation and delivery of suitable services during the life cycle of a consumer product is proposed in the first instance. The scheme consists of a product embedded Intelligent Data Unit (IDU), communication support infrastructure and a service enabler. The service enabler lies at the heart of the system. It is a collection of suitable software components capable of receiving product life cycle data and using it to provide integrated information and knowledge for the benefit of service providers.



The proposed service enabling scheme lays a foundation for the realisation of product-oriented PSS and use-oriented PSS for consumer products. Based on the general PSS definitions and the proposed service enabling scheme, a conceptual model of PSS for consumer products is devised. The conceptual model suggests a methodology for realising product-oriented PSS and use-oriented PSS for consumer products, where the service enabler is identified as a core building block. According to the methodology, each individual product should be embedded with an IDU to acquire the dynamic life cycle data. The acquired dynamic data and associated static data are then transferred to a central service enabler to generate integrated information and knowledge which can be used by service providers.

In order to provide a general reference for realising product-oriented PSS and use-oriented PSS for consumer products, a PSS reference architecture is formulated. It presents internal constructions for both PSS and the service enabler.

The development of the service enabler has been identified as the crucial challenging step during the realisation process. In order to facilitate the realisation of PSS, a major software toolset, namely, service enabler software framework has been devised. This has been referred to as the Framework in this thesis. This software framework is primarily implemented and delivered as a set of components, class libraries and APIs, dynamic data structure schema and life cycle database schema.

Two product-oriented PSS prototypes have been built during ELIMA field trials utilising the proposed PSS realisation methodology and the associated tools. It has been demonstrated via two product-oriented PSS prototypes that the delivery of services can be facilitated. The enabled services demonstrate that the proposed methodology and tools are appropriate and effective for realising product-oriented PSS for consumer products. Field trials also indicate that acquiring and processing the product life cycle data through appropriate Information and Communication Technologies (ICT) can also facilitate the delivery of other services such as product rental or sharing service. Hence the proposed methodology and tools are appropriate and effective for the practical realisation of use-oriented PSS, in addition to the product-oriented PSS.



Table of Contents

**DECLARATION.....I**

**ACKNOWLEDGEMENT .....II**

**SYNOPSIS .....III**

**TABLE OF CONTENTS .....V**

**LIST OF FIGURES .....XIII**

**LIST OF TABLES .....XVI**

**LIST OF ACRONYMS AND ABBREVIATIONS .....XVII**

**Chapter 1 Introduction .....1**

1.1 Background .....1

1.2 Problem Statement .....3

1.3 Research Aim and Objectives .....5

1.4 Research Approach.....5

1.5 Thesis Organisation.....8

**Chapter 2 Literature Survey I – Sustainable Product Development.....10**

2.1 Introduction .....10

2.2 Sustainable Development and Product Service Systems .....10

2.2.1 Historical Background of Sustainable Development .....10

2.2.2 Strategies towards Sustainability .....11

2.2.2.1 Product Side: eco-design.....12

2.2.2.2 Production Side: dematerialisation and servicification .....13

2.2.2.3 Consumption Side: renting, leasing and sharing .....14

2.2.3 Product Service Systems (PSS) .....15

2.2.3.1 PSS Concept .....15

2.2.3.2 Product-oriented PSS .....16

2.2.3.3 Use-oriented PSS .....16

2.2.3.4 Result-oriented PSS .....16

2.2.4 A Brief History of PSS Research and Identified Problems.....17

2.2.5 Policy Implications .....18

2.2.5.1 WEEE and RoHS Directives.....19



2.2.5.2	Eco-design Requirements for Energy-using Products (EuP) .....	20
2.2.5.3	Implications .....	20
2.2.6	Summary .....	21
2.3	Engineering Tools and Methods for PSS .....	22
2.3.1	Life Cycle Engineering: Life Cycle Oriented PSS Design .....	22
2.3.2	Service Engineering: Product-Service Co-Design and Service Modelling.....	24
2.3.3	Software Engineering: Scenarios and Use Cases.....	25
2.3.4	Service CAD .....	27
2.3.5	New Service Development .....	27
2.3.6	MEPSS.....	29
2.3.7	Summary .....	30
2.4	Product Life Cycle Data and Acquisition.....	31
2.4.1	Product Life Cycle.....	31
2.4.1.1	Beginning-of-Life (BOL).....	31
2.4.1.2	Middle-of-Life (MOL).....	31
2.4.1.3	End-of-Life (EOL).....	33
2.4.2	Product Life Cycle Data.....	33
2.4.2.1	Static Data.....	33
2.4.2.2	Dynamic Data .....	34
2.4.3	Product Life Cycle Data Acquisition.....	35
2.4.3.1	Intelligent Data Unit (IDU).....	35
2.4.3.2	“Whitebox” Project.....	36
2.4.4	Summary .....	37
2.5	ELIMA Project .....	38
2.5.1	ELIMA Background .....	38
2.5.2	ELIMA Field Trials .....	39
2.5.3	Research Study in Association with ELIMA.....	40
2.6	Chapter Summary.....	41
 <b>Chapter 3 Literature Survey II - Enabling Technologies for Product Life</b>		
<b>Cycle Information Management .....</b>		<b>43</b>
3.1	Introduction .....	43
3.2	Database and Information Management System.....	43
3.2.1	Database and Database Management System (DBMS).....	44
3.2.2	History of Database and DBMS .....	45



3.2.3	Database Design: E-R Modelling and Normalisation.....	46
3.3	Information System Development and Software Framework.....	48
3.3.1	Information System Development Approaches .....	48
3.3.2	Object-Oriented Analysis and Design.....	52
3.3.3	Unified Modelling Language (UML) .....	54
3.3.4	Software Framework.....	55
3.4	XML and Web Services .....	56
3.4.1	XML: Extensible Markup Language .....	56
3.4.2	Web Services .....	57
3.5	Web Technologies.....	58
3.5.1	Java /J2EE.....	58
3.5.2	Microsoft .NET Platform.....	60
3.5.3	PHP .....	61
3.6	State-of-Art Automatic Identification (Auto-ID) Technologies .....	62
3.6.1	Radio Frequency Identification (RFID) Technology.....	62
3.6.2	Electronic Product Code (EPC) .....	64
3.6.3	Product Markup Language (PML).....	65
3.6.4	Object Naming Services (ONS).....	66
3.7	Summary .....	66

## **Chapter 4 A Scheme of Enabling Services from Product Life Cycle**

Data.....	68
4.1 Introduction .....	68
4.2 Service and Service Content.....	69
4.2.1 Service Definition.....	69
4.2.2 Service Content.....	69
4.3 Rationale of Enabling Services from Product Life Cycle Data.....	69
4.3.1 The Value of Product Life Cycle Data.....	70
4.3.2 Knowledge Discovery.....	70
4.3.3 Data, Information and Knowledge Can Enable Services.....	72
4.4 Service Enabling.....	73
4.4.1 Service Enabling Scheme .....	73
4.4.2 Service Enabler .....	74
4.4.3 Service Enabling Process.....	75



4.5 Summary .....77

**Chapter 5 The Realisation of Product Service Systems for Consumer Products.....78**

5.1 Introduction .....78

5.2 PSS Requirements Analysis .....79

5.2.1 PSS Requirements Elicitation .....79

5.2.2 PSS Requirements Specification.....79

5.3 PSS Conceptual Model and Realisation Methodology .....80

5.4 A Typical Implementation of PSS for Consumer Products .....82

5.5 PSS Architecture .....83

5.5.1 PSS Architecture Overview .....84

5.5.2 Service Enabler Internal Structure .....85

5.5.2.1 Life Cycle Data Communication Interface .....85

5.5.2.2 Service Content Delivery Interface .....85

5.5.2.3 Security Manager .....86

5.5.2.4 Data/Information Store .....86

5.5.2.5 Information Engine .....87

5.5.2.6 Knowledge Base and Inference Engine .....90

5.5.2.7 Service Enabler Manager .....91

5.6 Relationship between the Proposed Methodology and ELIMA Project .....92

5.7 Summary .....93

**Chapter 6 Analysis and Design of the Framework.....95**

6.1 Introduction .....95

6.2 IMS Use Case Global View .....96

6.3 IMS Three-tier Distributed Architecture .....98

6.3.1 Information System Three-tier Architecture.....98

6.3.2 Three-tier Architecture of IMS in ELIMA .....98

6.4 The Framework Architectural Design.....100

6.5 Life Cycle Data Design and Handling.....101

6.5.1 Static Data Handling.....102

6.5.2 Dynamic Data Handling .....103

6.5.3 Development of Generic Life Cycle Data Structure Schema .....105

6.5.3.1 Dynamic Data Acquisition Modes .....105



---

6.5.3.2	Generic Dynamic Data Structure Schema.....	106
6.6	Database Design .....	108
6.6.1	Product Life Cycle Database Specification .....	108
6.6.2	E-R Modelling Diagram .....	109
6.6.3	Life Cycle Database Schema .....	110
6.7	Class Libraries and APIs .....	111
6.8	Components Identification and Categorisation .....	112
6.9	The Design of Communication Manager .....	114
6.9.1	Neutral Format Layer.....	114
6.9.2	File - based Data Transfer.....	115
6.9.3	Live Data Transfer .....	116
6.10	The Design of Security Manager.....	117
6.10.1	Three-level Security Control .....	118
6.10.2	Role and Session Paradigm .....	119
6.11	The Design of Information Engine.....	120
6.11.1	On-line Data Analysis .....	120
6.11.2	Transformation of Data into Excel Spreadsheet.....	122
6.12	The Design of Query Engine.....	123
6.13	The Design of Scheduler .....	124
6.14	The Design of Notifier .....	124
6.15	The Design of Report Generator .....	124
6.16	The Design of Product Registration .....	125
6.17	Summary .....	126
<b>Chapter 7 Implementation of the Framework and Supporting Tools....</b>		<b>128</b>
7.1	Introduction .....	128
7.2	Development Tools and Implementation Strategy.....	128
7.2.1	Development Tools.....	128
7.2.2	Implementation Strategy.....	129
7.3	The Framework Implementation .....	129
7.3.1	Runtime Infrastructure.....	129
7.3.2	Information Engine .....	132
7.3.2.1	Excel Spreadsheet Applet .....	132
7.3.2.2	Java-JavaScript Live Connection.....	134



7.3.2.3	Provision of on-line analysis toolkit .....	135
7.3.3	Report Generator.....	136
7.3.3.1	Chart Generation.....	136
7.3.3.2	Excel Spreadsheet Generation.....	136
7.3.4	File Upload Applet.....	137
7.3.5	Scheduler .....	138
7.3.6	Query Engine .....	139
7.3.7	Security Manager.....	140
7.3.8	Notifier.....	140
7.3.9	Database Administration.....	141
7.3.10	Product Registration .....	142
7.4	Supporting Tools .....	143
7.5	Summary .....	143
 <b>Chapter 8 Using the Framework to Realise a Service Enabler Solution.</b>		<b>144</b>
8.1	Introduction .....	144
8.2	Plug Points.....	145
8.3	Service Enabler Realisation Process .....	146
8.4	Summary .....	147
 <b>Chapter 9 Test Cases .....</b>		<b>148</b>
9.1	Introduction .....	148
9.2	Test Case Design .....	148
9.2.1	Test Scope.....	149
9.2.2	Test Approach .....	149
9.2.3	Expected Results and Test Criteria .....	150
9.2.4	Assumptions and Constraints.....	150
9.3	Test Case I: Fridge/freezer PSS Prototype .....	151
9.3.1	ELIMA White Goods Field Trial Overview .....	151
9.3.1.1	Trial Product and Specification.....	151
9.3.1.2	IDU and Data Acquisition.....	153
9.3.2	Using the Framework to Create FFIMS.....	155
9.3.3	FFIMS Evaluation.....	157
9.3.3.1	User Interface Overview .....	157
9.3.3.2	Security Management .....	158

9.3.3.3 Data Communication: File Upload ..... 159

9.3.3.4 Data Analysis ..... 160

9.3.3.5 Evaluation Result ..... 162

9.3.4 Demonstration of Enabled Services..... 163

9.3.4.1 Scenario 1 - Remote Monitoring..... 163

9.3.4.2 Scenario 2 - End-of-Life Treatment..... 165

9.3.5 Summary of Test Case I..... 167

9.4 Test Case II: PlayStation 2 PSS Prototype..... 167

9.4.1 ELIMA Brown Goods Field Trial Overview ..... 167

9.4.1.1 Trial Product and Specification..... 167

9.4.1.2 IDU and Data Acquisition..... 168

9.4.1.3 Data Flow..... 171

9.4.2 Using the Framework to create PSIMS..... 172

9.4.2.1 Algorithm to Extract and Store Standard Event Data ..... 173

9.4.2.2 Algorithm to Determine the Standard Event Dropout(s) ..... 174

9.4.2.3 Algorithm to Parse the XML Document..... 176

9.4.3 Demonstration of Enabled Services..... 178

9.4.3.1 Scenario 1 - Remote Diagnostics ..... 178

9.4.3.2 Scenario 2 - End-of-Life Treatment..... 179

9.4.3.3 Scenario 3 - Use Pattern Analysis..... 181

9.4.4 Summary of Test Case II ..... 182

9.5 Appraisal of Test Results ..... 183

9.6 Summary ..... 185

**Chapter 10 Conclusions and Future Work ..... 187**

10.1 Conclusions from the Research Study..... 187

10.2 Contribution to Knowledge ..... 189

10.3 Recommendations and Future Work..... 190

**References ..... 192**



**APPENDIX A: Class Libraries and APIs ..... 206**

    A.1 Class and Libraries in Java .....206

    A.2 Class and Libraries in PHP .....217

**APPENDIX B: Supporting Tools .....232**

**APPENDIX C: List of Publications .....233**

List of Figures

Figure 2.1 Service design process model .....22

Figure 2.2 Life cycle oriented PSS design (Aurich et al, 2004).....23

Figure 2.3 The NSD process cycle (Fitzsimmons et al, 2000).....28

Figure 2.4 An example of use pattern analysis.....32

Figure 2.5 IDU reference architecture (Moore et al, 2000).....37

Figure 2.6 Product, data and information flow in ELIMA (ELIMA, 2004) .....38

Figure 2.7 ELIMA field trials overview.....39

Figure 3.1 Database and DBMS (Rolland, 1998) .....44

Figure 3.2 Unified software development process (Jacobson et al, 1999).....51

Figure 3.3 J2EE architecture (J2EE, 2004) .....59

Figure 3.4 A basic RFID system (Finkenzeller, 2003).....64

Figure 4.1 Service enabling scheme conceptual diagram .....68

Figure 4.2 Data, information and knowledge transformation .....71

Figure 4.3 Schematic presentation of service enabling scheme.....74

Figure 4.4 Service enabling process.....76

Figure 5.1 Development process of PSS realisation methodology .....78

Figure 5.2 PSS requirements elicitation .....79

Figure 5.3 A conceptual model for product-oriented PSS and use-oriented PSS .....80

Figure 5.4 A typical implementation of PSS for consumer products.....82

Figure 5.5 An architecture for product-oriented PSS and use-oriented PSS.....84

Figure 5.6 Database management system (Rolland, 1998).....86

Figure 5.7 Information engine modules .....87

Figure 5.8 Data mining (Thearling,2004) .....88

Figure 5.9 Consuming Web services.....89

Figure 5.10 Expert system .....90

Figure 5.11 Relationship between research study and ELIMA project.....93

Figure 6.1 Development and test plan.....96

Figure 6.2 Use case global view of IMS in ELIMA .....97

Figure 6.3 Three-tier distributed architecture (Middleware, 2004).....98

Figure 6.4 IMS three- tier distributed architecture.....99



Figure 6.5 The Framework model .....100

Figure 6.6 A general static data handling.....102

Figure 6.7 General dynamic data handling .....103

Figure 6.8 Dynamic data structure in field trials.....106

Figure 6.9 Dynamic life cycle data structure description.....107

Figure 6.10 Dynamic life cycle data structure schema.....108

Figure 6.11 Life cycle database E-R diagram .....110

Figure 6.12 Components hierarchy .....113

Figure 6.13 Communication manager neutral format layer .....114

Figure 6.14 Sequence diagram of file upload operation.....116

Figure 6.15 A solution for live data transfer .....117

Figure 6.16 Three-level access control.....118

Figure 6.17 Service enabler security model .....119

Figure 6.18 Data analysis in information engine .....120

Figure 6.19 On-line data analysis.....121

Figure 6.20 Transformations into Excel spreadsheet .....122

Figure 6.21 Task scheduling .....124

Figure 6.22 Report generator.....125

Figure 7.1 Runtime infrastructures within the Framework .....129

Figure 7.2 A screenshot of spreadsheet-based data display Applet .....133

Figure 7.3 The process of on-line data analysis .....134

Figure 7.4 Live connect (Flannagan, 1997) .....135

Figure 7.5 Reuse of Excel worksheet functions .....135

Figure 7.6 Excel spreadsheet generation.....137

Figure 7.7 Screenshot of HTML file upload interface .....137

Figure 7.8 A screenshot of using JUpload Applet for file upload.....138

Figure 7.9 Microsoft task scheduler in control panel.....139

Figure 7.10 Sequence diagram of query operation.....139

Figure 7.11 Operation sequence of security model .....140

Figure 7.12 A screenshot of PhpMyAdmin.....141

Figure 7.13 Product registration.....142

Figure 8.1 The process of using the Framework to realise a service enabler solution..144

Figure 8.2 The software framework instantiation .....145

Figure 9.1 Test approach .....149

Figure 9.2 Product-oriented PSS prototypes built in ELIMA project.....150

Figure 9.3 Trial product: fridge/freezer.....151

Figure 9.4 Data transmission through GSM/PSTN modem.....154

Figure 9.5 Support GSM/PSTN modem data transmission.....155

Figure 9.6 Schedule the task.....156

Figure 9.7 A snapshot of data extraction summary report .....156

Figure 9.8 A screenshot of the FFIMS interface .....158

Figure 9.9 A screenshot of creating a user account.....158

Figure 9.10 A screenshot of file upload .....159

Figure 9.11 Snapshot of dynamic data display in HTML table .....160

Figure 9.12 The export of dynamic data to Excel spreadsheet .....161

Figure 9.13 A screenshot of on-line analysis: histogram .....161

Figure 9.14 A screenshot of on-line analysis result: correlation.....162

Figure 9.15 A screenshot of freezer monitoring information.....163

Figure 9.16 Compressor cycles v fridge door openings (ELIMA Seminar, 2005) .....164

Figure 9.17 Access BoM through FFIMS .....165

Figure 9.18 A screenshot of BoM for COM 95.....166

Figure 9.19 Trial product: PlayStation2 .....168

Figure 9.20 PlayStation2 IDU specification (ELIMA D25, 2004) .....169

Figure 9.21 Brown goods field trial data flow (ELIMA Seminar, 2005).....172

Figure 9.22 Data extraction algorithm .....174

Figure 9.23 Algorithm to determine the number of standard event dropout(s) .....175

Figure 9.24 Algorithm of an XML parser for PS2 data readout .....177

Figure 9.25 Remote diagnostics: A data display screenshot from PSIMS.....178

Figure 9.26 A screenshot of PS2 end-of-life treatment information.....180

Figure 9.27 Preferred operation modes of PS2 consoles (ELIMA D25, 2004) .....182



List of Tables

Table 3.1 Comparison between J2EE and .Net .....60

Table 9.1 PS2 life cycle data to be recorded .....170

Table 9.2 PS2 standard event data to be recorded .....170

Table 9.3 PS2 dramatic event data to be recorded .....171

## List of Acronyms and Abbreviations

3D	Three Dimensional
ANSI	American National Standard Institute
AOCS	Attitude and Orbit Control Subsystem
APC	Antenna Pointing Control
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
ASP	Active Server Pages
Auto-ID	Automatic Identification
B2B	Business to Business
B2C	Business to Customer
BMI	Body Mass Index
BOL	Beginning-of-Life
BoM	Bill of Materials
CAD / CAM	Computer Aided Design / Computer Aided Manufacturing
CARE	Comprehensive Approach for Recycling of Electronics
CFML	Cold Fusion Markup Language
COM	Component Object Model
CODASYL	Conference On Data Systems Languages
CSI	Communication Support Infrastructure
CSV	Common Separated Value
DBMS	Database Management System
DBTG	Database Task Group
DDL	Data Definition Language
DML	Data Manipulation Language
DMU	De Montfort University
DNA	Distributed interNet application Architecture
DNS	Domain Name System
DOM	Document Object Model
DTD	Document Type Definition
DTI	Department of Trade and Industry
EC	European Commission



EIS	Enterprise Information System
EJB	Enterprise Java Beans
ELIMA	Environmental Lifecycle Information Management and Acquisition
EOL	End-of-Life
EPC	Electronic Product Code
EPSRC	Engineering and Physics Science Research Council
E-R	Entity-Relationship
EU	European Union
EuP	Energy-using Product
FFIMS	Fridge/Freezer Information Management System
FTP	File Transfer Protocol
GSM	Global System for Mobile Communications
GUAM	Generalised Update Access Method
GUI	Graphic User Interface
HTML	HyperText Markup language
HTTP	HyperText Transfer Protocol
ICT	Information and Communication Technology
IDE	Integrated Design Environment
IDEF	Integration Definition for Function Modelling
IDU	Intelligent Data Unit
IID	Iterative and Increment Development
IMAS	Intelligent Machines and Automation System
IPP	Integrated Product Policy
IPPM	Integrated Product-Process Model
J2EE	Java 2 Enterprise Edition
JRE	Java Runtime Environment
JSP	Java Server Pages
JVM	Java Virtual Machine
KDD	Knowledge Discovery in Database
LAN	Local Area Network
LCA	Life Cycle Analysis
LCE	Life Cycle Engineering
MEPSS	MEthodology for Product Service Systems

<b>MOL</b>	<b>Middle-of-Life</b>
<b>MSMQ</b>	<b>MicroSoft Message Queue</b>
<b>MTS</b>	<b>Microsoft Message Server</b>
<b>NAA</b>	<b>North American Aviation</b>
<b>NPD</b>	<b>New Product Development</b>
<b>NSD</b>	<b>New Service Development</b>
<b>OMG</b>	<b>Object Management Group</b>
<b>OMT</b>	<b>Object Modelling Techniques</b>
<b>ONS</b>	<b>Object Naming Service</b>
<b>OO</b>	<b>Object Oriented</b>
<b>OODBMS</b>	<b>Object Oriented DBMS</b>
<b>ORDBMS</b>	<b>Object Relational DBMS</b>
<b>PBB</b>	<b>Poly-Brominated Biphenyl</b>
<b>PBDE</b>	<b>Poly-Brominated Diphenyl Ethers</b>
<b>PHP</b>	<b>PHP: Hypertext Preprocessor</b>
<b>PI-NET</b>	<b>Product service system information NETwork</b>
<b>PLM</b>	<b>Product Lifecycle Management</b>
<b>PML</b>	<b>Product Markup Language</b>
<b>PMPP</b>	<b>Post Mass Production Paradigm</b>
<b>PS2</b>	<b>Play Station2</b>
<b>PSIMS</b>	<b>Play Station2 Information Management System</b>
<b>PSS</b>	<b>Product Service Systems</b>
<b>RDD</b>	<b>Responsibility-Driven Design</b>
<b>RDBMS</b>	<b>Relation Database Management System</b>
<b>RF</b>	<b>Radio Frequency</b>
<b>RFID</b>	<b>Radio Frequency Identification</b>
<b>R&amp;O</b>	<b>Research &amp; Development</b>
<b>RoHS</b>	<b>Restrictive of Hazardous Substance</b>
<b>RTOS</b>	<b>Real Time Operating System</b>
<b>SME</b>	<b>Small Manufacturer Enterprise</b>
<b>SAX</b>	<b>Simple API for XML</b>
<b>SGML</b>	<b>Structured Generalised Markup Language</b>
<b>SKU</b>	<b>Stock Keeping Unit</b>



SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SusProNet	Sustainable Product Service Network
TCP/IP	Transmission Control Protocol / Internet Protocol
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
UNCED	United Nation Conference on Environment and Development
UNEP	United Nations Environment Programme
USDP	Unified Software Development Process
W3C	World Wide Web Consortium
WCED	World Commission on Environment and Development
WSDL	Web Service Description Language
WEEE	Waste Electrical and Electronic Equipment
XML	eXtensible Markup Language
XPath	XML Path Language
XSL	eXtensible Stylesheet Language
XSLT	eXtensible Stylesheet Language Transformation

# Chapter 1 Introduction

This research study aims to investigate engineering methods and tools for realising Product Service Systems (PSS) for typical consumer products. During the last two decades, a very fast growth in the consumer product market is obvious. However, challenging factors such as high customer expectation, globalisation complexities, social/economical/environmental sustainability issues and safety/legislation/ethical aspects have to be faced while maintaining good human practice. Therefore the research work in relation to PSS is not only timely but also essential.

This research proposes a methodology which can effectively utilise product life cycle data to improve and open new horizons for the design and service of consumer products. The study has the potential to create product design/service methodologies, which are sustainable, efficient in the consumption of resources, and are environmentally sensitive. Thus, it will contribute to the exploration of eco-design and development of next generation consumer products (e.g. smart home appliances) in the context of good business practice.

This chapter presents an overview of the work covered in the project.

## 1.1 Background

Sustainable development advocates reduction of resource consumption, while delivering better and more widely available goods and services. Therefore strategy for developing PSS for social, economical and environmental sustainability is becoming increasingly important (UNEP & PSS, 2001; SusProNet, 2004; Wong, 2003; Mont, 2000; Manzini et al 2002). PSS are commonly defined as “a marketable set of products and services capable of jointly fulfilling a consumer need” (Goedkoop et al, 1999). PSS aim to replace conventional products with integrated product services in an attempt to get better use of resources, improve economic performance, reduce the overall environmental impact and enhance the well being and satisfaction of the consumer to achieve the sustainability. It means a change of focus from traditional production



oriented philosophy which concentrates on intensive use of materials and energy, to goods where service content is dominant and partially replaces material intensive production (Mont, 2000; Manzini et al, 2002; Tomiyama et al, 2000; Wong, 2003).

Design, development and deployment of PSS are intrinsically complex in nature (Tomiyama, 2003). Any available technological support would benefit the situation, but so far, only limited attention has been given to methodologies and tools for the realisation of PSS (Tomiyama, 2003; Wong, 2003; Mont, 2003; Van der Horst et al, 1999). In order to address this need, some engineering methods and tools, such as service engineering, life cycle engineering, “use case & scenario” methods from software engineering and MEPPS (Methodology for Product Service Systems) toolkit, are used or developed to seek the design, development, implementation and deployment of PSS.

The discipline of service engineering, which was first proposed in the mid 90’s in Germany and Israel (Bullinger et al, 2003; Mandelbaum et al, 1998), is concerned with the systematic development of services using suitable models, methods and tools. Service engineering promotes an integrated service within a product life cycle by adopting technological methods and employ existing engineering know-how to maximise efficiency (Tomiyama, 2001; Bullinger et al, 2003). Broadly speaking, service engineering includes methods of product service co-design (Bullinger et al, 2003; Mandelbaum et al, 1998 and Ganz et al, 2004), service modelling (Tomiyama, 2001), service CAD (Tomiyama, 2003) and New Service Development (NSD) (Fitzsimmons et al, 2000). These service engineering methods seek the design, development, and production of services and aim at exploring ways for the realisation of PSS.

Life cycle-oriented PSS design (Aurich et al, 2004) adopts Life Cycle Engineering (Jeswiet et al, 2003) (LCE), which is usually used in the context of physical product design, for service design and PSS design. LCE emerged in 2003, and is defined as “design and manufacture of products with the goal of protecting the environment and

conserving resources, while encouraging economic progress, and at the same time optimising the product life cycle”.

Morelli (2002) borrowed the idea of “scenario and use case” from software engineering for the analysis, design and development of PSS, since “use cases and scenarios” are useful analysis and design methods commonly used in the object oriented software to help design and develop information management system.

Similarly, the importance of this sphere of work is exemplified by a European Commission funded project called Methodology for Product Service Systems (MEPSS). MEPSS aims to provide industry with a toolkit that enables them to analyse newly developed product-services with regard to design and implementation aspects, economical impacts, social and environmental impacts and issues related to consumer acceptance and culture and ethics (Pi-Net, 2002).

## 1.2 Problem Statement

Problems identified through the review of engineering methods and tools for PSS are described as follows:

- (1) Present engineering methods and tools for PSS have not fully resolved the issue of how an integrated mechanism can enable suitable services to be delivered effectively and efficiently during a product life cycle.

PSS promotes that products should have greater added value derived from knowledge and content rather than from the monetary value of raw materials, thus the cost of materials and energy can be reduced to assist sustainability (Tomiya, 2001; Manzini et al, 2002). It hence follows that an integrated mechanism to enable suitable services which are proactive during the life cycle of a product is crucial for the whole PSS development process. However, present engineering methods and tools for PSS have not fully resolved this issue. They are mainly concerned with the general design and development of service and PSS. Service engineering method and tools (i.e. product



service co-design, service modelling, service CAD, NSD) highlight design and development of services; life cycle oriented PSS design and “scenario and use case” based design are focused on the general design and development of PSS. MEPSS project is carried out to provide methods and tools to analyse the newly developed product-services. Very few of them have addressed the issue of how an integrated mechanism can enable services to be delivered effectively and efficiently.

(2) Present engineering methods and tools for PSS are unlikely to help industrial manufacturers practically implement and realise sustainable PSS solutions.

Research into developing practical tools to help manufacturer implement sustainable PSS solutions is still at a formative stage (Wong, 2003). In the domain of consumer goods, lack of dissemination of information about successful PSS solutions directly results in insufficient know-how about PSS in consumer goods sector (Mont, 2003; Wong, 2003). From industrial manufactures’ perspective, they need engineering methods and tools with predictive power that can help them practically implement and realise a sustainable PSS solution (Wong, 2003; Mont, 2003; Van der Horst, 1999; Tomiyama, 2003).

Considering problems outlined above, there is an urgent need to seek innovative engineering tools and methods to realise services and PSS. In order to address these issues, an investigation of engineering methods and tools for the realisation of PSS for consumer products has been carried out in association with the European Commissions Framework V funded project entitled Environmental Life-cycle Information Management and Acquisition for consumer products(ELIMA) (<http://www.elima.org> ).

### 1.3 Research Aim and Objectives

**Aim:**

The aim of this research study is to investigate engineering methods and tools for realising PSS solution for consumer products. The aim can be further broken down into the following objectives.

**Objectives:**

- 1) To study and devise engineering methods for realising PSS solution for consumer products.
- 2) To formulate an associated toolkit i.e. the Framework to facilitate the realisation of PSS for consumer products.
- 3) To create two prototype service enablers, namely, Fridge/freezer Information Management System (FFIMS) and PlayStation2 Information Management System (PSIMS) by using the Framework in the context of PSS design.
- 4) To verify the proposed methodology and the associated toolkit, i.e. Framework. The verification process will be based on the appropriateness and effectiveness of the proposed methodology and associated toolkit through the delivery of identified services.

### 1.4 Research Approach

The approach of the research study is through the following 4 steps:

1. Propose a scheme which actively utilises product life cycle data in an integrated and systematic way to help creation and delivery of effective services during the life cycle of a product.
2. Develop a methodology for realising product-oriented PSS and use-oriented PSS solution for consumer products based on the proposed service enabling scheme.



3. Develop the Framework and associated tools to create product-specific service enabler.
4. Verify the proposed scheme, methodology and toolkit during ELIMA field trials.

Step 1: Propose a scheme which actively utilises product life cycle data in a systematic and integrated way to help creation and delivery of effective services during the life cycle of a product.

The proposed service enabling scheme relies on:

- (i) A product embedded with an Intelligent Data Unit (IDU) (Moore et al, 2000; Simon et al, 1998; Simon et al, 2000) which is responsible for acquiring product life cycle data. Each individual product can be equipped with IDU capable of the storage and transmission product life cycle data.
- (ii) Communication Support Infrastructure (CSI) which provides a communication channel for transmission of life cycle data and service content.
- (iii) A service enabler which is a collection of suitable software components such as database, knowledge base, information engine and inference engine, etc capable of receiving product life cycle data and using it to provide integrated information and knowledge which can be used by service providers.

Step 2: Propose a methodology for the realisation of product-oriented and use-oriented PSS for consumer products

The proposed service enabling scheme lays a foundation for the realisation of product-oriented PSS and use-oriented PSS for consumer products. Based on the general PSS definitions and the proposed service enabling scheme, a conceptual model is devised. This conceptual model suggests an approach for realising appropriate PSS for consumer products. In this solution, the service enabler is identified as a core building block.

Architecture for PSS is developed based on the PSS conceptual model. The architecture not only presents internal constructions for both PSS and component modules (e.g. service enabler), but also the interactions amongst those component modules and sub-modules.

### Step 3: Develop a software framework and associated supporting tools

The development of the service enabler has been identified as the crucial challenging procedure during the realisation process. In order to facilitate the realisation of PSS, a toolkit i.e. the Framework and related supporting tools are developed in association with ELIMA project.

The Framework is primarily implemented and delivered as a set of components, class libraries and APIs (Application Programming Interface), dynamic data structure schema and life cycle database schema. Analytical methodology (e.g. “use cases” and “scenarios”), modelling language (i.e. UML - Unified Modelling Language, database E-R modelling – Entity-Relationship modelling), Web technologies (e.g. Java, XML – eXtensible Markup Languages) and software (i.e. Rational Rose, Macromedia Dreamweaver MX) have been employed to facilitate both analysis and design processes. The Framework has also been designed to include a typical service enabler runtime infrastructure which is built by employing a variety of off-the-shelf products or technologies.

### Step 4: Verify the proposed scheme, methodology and tools in ELIMA field trials

Test the proposed scheme, methodology and tools via two product-oriented PSS prototypes built utilising the proposed methodology.



## 1.5 Thesis Organisation

The thesis is organised in 10 chapters including an **Introduction** and **Conclusions**. An overview structure of the thesis is outlined as follows:

**Chapter 2:** (Literature Survey I – Sustainable Product Development) reviews the literature related to sustainable development and PSS. The review focuses on: (i) the sustainability and PSS (ii) engineering methods and tools for PSS (iii) product life cycle data acquisition (iv) ELIMA project.

**Chapter 3:** (Literature Survey II – Enabling Technologies for Product Life Cycle Information Management) reviews the literature related to enabling technologies for product life cycle information management. Five topics of enabling technology have been reviewed, namely, (i) database and information management system, (ii) information system development and software framework, (iii) eXtensible Mark-up Language (XML) and Web Services, (vi) Web technologies, and (v) state-of-art Automatic Identification (Auto-ID) technologies.

**Chapter 4:** (A Scheme of Enabling Services from Product Life Cycle Data) proposes a scheme of enabling services from product life cycle data.

**Chapter 5:** (PSS Realisation Methodology for Consumer Products) proposes a PSS realisation methodology for consumer products based on the proposed service enabling scheme. A conceptual model which suggests an approach for realising PSS for consumer products and a PSS architecture are devised.

**Chapter 6:** (Analysis and Design of the Framework) describes the analysis and design of the Framework to facilitate the creation of service enabler. This Framework is designed to include a run time infrastructure, components, class libraries/APIs and schemas.

**Chapter 7:** (Implementation of the Framework and Supporting Tools) describes implementation of the Framework and related supporting tools.

**Chapter 8:** (Using the Framework to Realise a Service Enabler Solution) introduces how to use the Framework to create a product-specific service enabler.

**Chapter 9:** (Test Cases) describes two test cases designed to test the proposed scheme, methodology and tools.

**Chapter 10:** (Conclusions and Future Work) summaries conclusions from the research study, contribution to knowledge as well as recommendations and future work.



# Chapter 2 Literature Survey I – Sustainable Product Development

## 2.1 Introduction

Investigating methodologies and tools for realisation of Product Service Systems (PSS) for consumer products towards sustainability is the main goal within this research study. In this regard, this chapter reviews the literature related to sustainable development & PSS, engineering methods & tools for PSS and product life cycle data & acquisition. This chapter includes 4 major topics, namely, *Sustainable Development and Product Service Systems*, *Engineering Methods and Tools for PSS*, *Product Life Cycle Data and Acquisition* and *ELIMA Project*.

## 2.2 Sustainable Development and Product Service Systems

The objectives of sustainable development include fast and reliable customer service, prudent use of natural resources, sensitivity to environment, all in the context of social progress and economic growth (UK Government Sustainable Development, 2004). Thus the strategic importance of PSS in achieving sustainability is recognised as illustrated by many other authors (UNEP & PSS, 2001; SusProNet, 2004; Wong, 2003; Mont, 2000; Manzini et al, 2002).

### 2.2.1 Historical Background of Sustainable Development

Sustainable development was defined as “development that meets the needs of the present generation without compromising the needs for future generation” in the report “*Our Common Future*”, which was published in 1987 at the World Commission on Environment and Development (WCED) chaired by Norway prime minister Gro Harlem Brundtland. The international Brundtland Commission appointed by United Nations aims to propose strategies for improving human well-being in the short term without threatening environment in the long term (Brundtland, 1987). The commission advocates that economic development should attempt to decouple the link between resources consumption and economic growth.

The initial sustainable development concept presented at WCED stressed the role of industry and new production pattern. It claimed that product design and new production pattern should play vital role in achieving sustainability. It was proposed as a new development paradigm, which integrates economic growth, social development and environment protection as interdependent and mutually supportive elements for long-term development (Desai, 2002).

In 1992 the sustainable development concept was further developed at the United Nation Conference on Environment and Development (UNCED), the Rio Earth Summit held in June, 1992. UNCED stated that unsustainable pattern of consumption and production is the major cause of the continuous degradation of the global environment (UNCED, 1992). Same idea emerged in UNEP (1994) that sustainable development should include the need of new consumption patterns and consider the role of consumers.

Thus sustainable development requires that production and consumption systems should give more attention to value, human needs, natural resources, product and service functions (Van Weenen, 1996). In this perspective, sustainability means that both production and consumption activities can lead to economically, environmentally, and socially justified solutions. Hence sustainability can be realised by enhancing product design, and changing production and consumption pattern (UNEP & PSS, 2001; SusProNet, 2004; Mont, 2000).

### **2.2.2 Strategies towards Sustainability**

According to Brundtland (1987) the sustainability challenge consists of finding ways to sustain the provision of goods and services in such way that “meet the needs of the present without comprising the ability of future generation to meet their own needs”. Sustainable development features by economic growth based on social justness and reduction in the use of the natural resources. These three requirements should be attained harmoniously. Mont (2000) summaries the following approaches and strategies which can be employed to reduce the resource intensity of products and services:



- Dematerialisation
- Extending the product life
- Eco-efficiency
- Recycling and claiming the product material back
- Reducing requirement for product
- Increasing efficiency of the product usage phase

These suggested approaches towards sustainability can be divided into the following categories:

- Product Side: eco-design to improve environmental profile of a product for itself
- Production Side: dematerialisation and servicification (Votta, 2001) to add value largely from knowledge and service content instead of just materialistic value (Tomiyaama, 2000)
- Consumption Side: renting, leasing and sharing

These three approaches for sustainability are described as follows.

#### 2.2.2.1 Product Side: eco-design

Product eco-design addresses the relation between a product and the environment (UNEP, 2001). The ultimate goal of eco-design is to develop an environment-friendly product. Product eco-design can improve product's environmental performance in the following ways (Fiksel, 1996):

- *Material substitution*: replacing product constituents with substitute materials that are superior in terms of improved recyclability, reduce energy content, etc.
- *Waste source reduction*: reducing the mass of a product or its packaging.
- *Substance use reduction*: reducing or eliminating undesirable substances that either incorporated into a product or used in its manufacturing process.
- *Energy use reduction*: reducing the energy required to produce, transport, store, use, maintain, recycle, or dispose product and its packaging.
- *Life extension*: prolonging a product or its components' life span, thus reduce the associated waste.

- *Design for disassembly*: simplifying product disassembly and material recovery by using techniques such as fastening component or colour-coded plastics.
- *Design for recyclability*: ensuring high recyclable content in product materials and maximum recycling at end-of-life.
- *Design for disposability*: ensuring non-recyclable materials and components can be disposed safely and efficiently.
- *Design for reusability*: enabling components of a product to be recovered, refurbished and reused.
- *Design for energy recovery*: extraction of energy from waste materials
- *Modular design*: designing a product in modules enabling quick repair or change of a module in case of failure.

Product eco-design is an important strategy for sustainability. It not only can improve the product environment performance, but can also lead to a sustainable consumption pattern as the eco-designed product can force consumers to use the product in an environment-friendly way.

#### 2.2.2.2 Production Side: dematerialisation and servicification

Dematerialisation and servicification are effective ways to generate added value during a product life cycle to compensate for the decrease of production volumes (Tomiyaama, 2001). The essence of dematerialisation is to decouple the economic growth from the consumption of energy and raw materials, and add more service content to a product. This is also known as servicification (Votta, 2001; Tomiyama, 2000).

A new production paradigm called Post Mass Production Paradigm (PMPP) was proposed by Tomiyama (Tomiyaama, 1997) to differentiate from the current mass production mode. Mass production produces cheaper products with good quality and contributes to economic value creation. But constraints and limitations from natural resources and energy, and the ability to absorb the waste have led to the obvious problem of making, using, and disposing of increasing number of products. The PMPP paradigm aims to reduce the production and consumption volume of products to an



adequate, manageable size and bring this volume into balance with natural and social constraints. PMPP is achieved by dematerialisation and servicification through compensating for the reduction of production volume caused by PMPP.

The concepts of dematerialisation, servicification and PMPP have shown that providing enhanced service content and knowledge to products while reducing the consumption of raw materials without compromising a product's functionality is an effective way for sustainability.

#### 2.2.2.3 Consumption Side: renting, leasing and sharing

Over consumption of the limited resources is one of the main features of unsustainability. But it is unrealistic to cut consumption to achieve sustainability. Few people would like to minimise the consumption level they enjoy today. Thus radical changes at the consumption level are needed by moving the demand for products and services towards a more sustainable consumption pattern. The sustainable consumption implies departure from values and standards which are directly linked to a continual increase in material and energy consumption (Manzini and Vezzoli, 2002).

In order to achieve the sustainability by changing consumption pattern, a new business model as “to have a utility instead of an ownership” emerged (Tukker, 2003; Schrader, 1999). This encourages people to buy the service of the product rather than buying the ownership. Renting, leasing and sharing (or pooling) have recently been promoted to minimise the consumption of certain products. This means one product is used by several people.

Car-sharing (Meijkamp et al, 1998; Meijkamp et al, 1994) is a very typical example. There are more than 40 organisations in Germany that promote car-sharing (e.g. Stattauto, Berlin). The new business model of leasing and renting that do not have a purchasing option have more ecological advantages. Another example that is often cited is that of copy machines. Two leading companies in the copier business, Océ and Rank Xerox, have changed their attitude towards environmental issues by selling the function of the copier instead of selling products (Océ: The environment).

Future consumption systems should more focus on product utilisation and functions rather than physical or material value. Through promoting buying function or utilities rather than product ownership, the way products are designed and offered is changed and a sustainable consumption pattern can be realised.

### **2.2.3 Product Service Systems (PSS)**

It has been realised that services can provide values with less environment impact without comprising customers' needs. Now progressive manufacturers tend to put less focus on producing product but more interest in adding value to a customer through the provision of services. Within this context, the concept of PSS was proposed to promote a focus shift from selling just products to selling functions through a mix of products and services while fulfilling the same consumer demands with less environmental impact (Manzini and Vezzoli, 2002; Mont, 2000; Tomiyama et al, 2000; Wong, 2003; Tukker, 2003). It is therefore sufficiently evident that PSS has to be the corner stone in the strategy for sustainable development.

#### **2.2.3.1 PSS Concept**

PSS is most commonly defined as “a marketable set of products and services capable of jointly fulfilling a consumer need” (Goedkoop et al, 1999). This definition is too general and does not include any explicit criterion for sustainability (Wong, 2003). Mont (2000) defined PSS as “a system of products, services, supporting networks and infrastructure that is designed to be competitive, satisfy customer needs and have a lower environmental impact than traditional business models”. Mont claims that PSS solutions should facilitate the shift from the individual production and consumption system to the one, in which products, services, supporting infrastructure, and necessary networks are integrated together to provide a certain quality of life to consumers and, at the same time, minimise the environmental impact of system.

PSS is considered as a useful and attractive approach for sustainability as it fits well into the criteria of strategies to achieve the sustainability in the perspective of product, production and consumption (Mont, 2000; UNEP & PSS, 2001; Manzini and Vezzoli,



2002;Tukker,2003). The key idea behind PSS is that consumers do not specifically demand products, but rather are seeking the utility these products and services provide (UNEP & PSS, 2001). According to a service classification by Hockerts(1999) and SusProNet PSS Conference (2003), i.e. product-oriented service, use-oriented service and result-oriented, PSS can be categorised into product-oriented PSS, use-oriented PSS and result-oriented PSS accordingly (Tukker, 2003; Tischner,2004).

#### 2.2.3.2 Product-oriented PSS

The main feature of product-oriented PSS is that the product is owned by the consumer, and services delivered are attached to the product itself (Tukker, 2003; Tischner, 2004; SusProNet, 2003). For example, by incorporating maintenance, distribution services, etc into product sales, more added value can be achieved in addition to product sales. This often happens in package deals where manufacturers not only sell products but also provide customers relevant services including planning, financing, installation, operation, maintenance, upgrading, and recycling. In this case, the product is considered as a means to deliver services (Tomiyama, 2000).

#### 2.2.3.3 Use-oriented PSS

The typical examples of the use-oriented PSS are product renting, leasing or sharing, for example, car-sharing systems (Meijkamp et al, 1998; Meijkamp et al, 1994). The manufacturer/provider no longer sells the products, but only its usage and functions (Tukker, 2003; Tischner, 2004; Suspronet, 2003). The expected eco-efficiency of these types of services is closely associated with the high usage intensity, which results in a reduction in the amount of products needed (Hockerts, 1999; Schrader, 1999).

#### 2.2.3.4 Result-oriented PSS

In result-oriented PSS, the product is substituted by service which is owned and run by the manufacturer/ service provider and the physical “product” may not be easily identified (Tukker, 2003; Tischner,2004; Suspronet, 2003), for example, the delivery of warmth instead of delivering heating device. Manufacturer /service provider therefore

has the motivation to intensify services and optimise product's operation to achieve the sustainability. The result-oriented PSS mean a shift from buying products to buying services, which have a potential to minimise the environmental impact.

In brief, the product-oriented PSS aim to provide more product related services; while both use-oriented PSS and result-oriented PSS promote a new business model - "to have utility instead of ownership" consumption pattern to achieve the sustainability.

#### **2.2.4 A Brief History of PSS Research and Identified Problems**

In terms of ideas, concepts or research, according to Hockerts and Weaver (Hockerts & Weaver, 2001), the field of PSS has passed through the typical path of most fields of industrial study, from initial theoretical conceptualisation to a maturity using knowledge and expertise based on acquired evidence. Wong (2003) summarised the PSS development history as follows:

- Conceptualisation of the idea (defining what PSS is and why it might be useful, typically based on anecdotal evidence, early attempts at integration with established theories and creation of basic typologies).
- Increased pan-European co-operation leading to empirical PSS data and case studies, etc.
- Maturity (Development of frameworks and roadmaps; creation of a structured process for PSS design). Most examples work at a purely technical level rather than at a systemic and functional level, making the outcome of these projects useful for users such as designers who can pose specific, well-defined questions for the tool.

Having reviewed a lot of literature regarding PSS, Wong (2003) in his "*Product Service Systems in Consumer Goods Industry*" points out that a great deal of theoretical research has been carried out to define what PSS might be and why it might be useful; a significant



amount of practical research has been carried out to collect and analyse real-world PSS applications. However, cases for car-sharing (Meijkamp et al, 1994; Meijkamp et al, 1998) and chemicals management services (Kauffman et al, 1997; White et al, 1999) have really investigated the PSS issue in great depth. Very few of these cases have considered consumer goods.

Wong (2003) also indicates that research into developing practical tools to help manufacturers realise sustainable PSS solutions is still at a formative stage. Same arguments can also be found from other research work. For example, Van der Horst et al (1999) and Mont (2003) indicate that there is very little integration between any of the available tools and the case studies that have been carried out. “The serious analysis of PSS is only just beginning” (Votta, 2001). Tomiyama (2003) in his “*Service CAD*” stated that design, development and deployment of PSS is intrinsically complex in nature, and any available technological support would benefit the situation, but so far, little attention has been given to methodologies and tools to design, develop and deploy PSS.

### 2.2.5 Policy Implications

The European Commission has proposed a series of directives towards the sustainable development. These directives include a proposal for a *Directive on Waste Electrical and Electronic Equipment* (WEEE), a proposal for a *Directive on the Restriction of Hazardous Substance* (RoHS) in electrical and electronic equipment (DTI update on WEEE & RoHS, 2005) and *Directive on the eco-design of Energy-using Products* (EuP) (EuP Directives, 2005). WEEE and RoHS require producers to take the responsibility for the collection, disposal and recovery of the end-of-life product as well as to restrict the use of hazardous substances. EuP requires more energy-efficient products to be produced. How to comply with these directives has become an urgent issue to many manufacturers and recyclers (Cambridge Workshop 2004).

### 2.2.5.1 WEEE and RoHS Directives

The WEEE Directive encourages and sets criteria for the collection, treatment, recycling and recovery of waste electrical and electronic equipment (WEEE & RoHS Directive – DTI Website). It makes producers responsible for financing most of these activities (producer responsibility). This will provide incentives to design electrical and electronic equipment in an environmental more friendly way, which takes waste management aspects fully into account. Consumers will be able to return their equipment free of charge. There are targets for recycling and recovery of materials and components from the separately collected waste.

The RoHS Directive facilitates the dismantling and recycling of waste electrical and electronic equipment by restricting the use of hazardous substances used in their manufacture. From July 2006, the use of lead, mercury, cadmium, hexavalent chromium, Poly-Brominated Biphenyls (PBBs) and Poly-Brominated Diphenyl Ethers (PBDEs) will be banned in new products. PBBs and PBDEs are flame retardants used in plastics. The ROHS Directive provides a small number of exempted processes where the use of these materials will be allowed to continue in certain applications until alternatives are found.

In short, WEEE and RoHS Directives aim to:

- Reduce the waste from electrical and electronic equipment
- Restrict the use of hazardous substances
- Increase recovery and recycling rates
- Improve environmental performance of all stakeholders involved in the product life cycle
- Encourage separate collection and treatment



### 2.2.5.2 Eco-design Requirements for Energy-using Products (EuP)

The EuP proposal does not introduce directly binding requirements for specific products, but does define conditions and criteria for setting through subsequent implementation measures, requirements regarding environment relevant characteristics (such as energy consumption) and allows them to be improved quickly and efficiently. Products that fulfil the requirements will benefit both business and consumers, by facilitating free movement of goods across the EU and by enhancing product quality and environmental protection. By encouraging manufacturers to design products with the environmental impacts in mind throughout their entire life cycle, the European Commission (EC) implements an Integrated Product Policy (IPP) (IPP, 2005) and accelerates the move towards improving the environmental performance of energy-using products.

### 2.2.5.3 Implications

The enforcement of directives will have implications to product manufacturers, component manufacturers, importers, retailers, local authorities and consumers. Manufacturers might have to take responsibilities to (i) finance the collection, treatment and recovery of WEEE from central collection points (ii) ensure recovery/recycling targets are met (iii) provide information on reuse and treatment for each new type of product (vi) provide data on sales to enable market share calculation(WEEE & RoHS Directive –DTI Website). EuP requires manufacturers to produce more energy-efficient products.

In order to comply with the WEEE and RoHS, manufacturers/recyclers need to know a range of information in respect of the components they use in their products. This information would include items such as material content, customer feedback about the products, details of manufacture/recycler, remaining life span, etc. However, the situation is obscure due to a number of issues (Cambridge Workshop, 2004):

- Lack of information at the point of refurbishment/recycling
- Difficult traceability of parts (e.g. due to upgrade)

- Parts from different suppliers
- Parts from non EU-member countries
- Cost issues of take back and collection

### 2.2.6 Summary

Review of the research literature in relation to *Sustainable Development and Product Service Systems* reveals that:

- 1) Sustainability can be realised by enhancing product design, and changing production and consumption pattern; developing PSS becomes an important strategy towards the sustainability.
- 2) PSS can be categorised into three types:
  - Product-oriented PSS
  - Use-oriented PSS
  - Result-oriented PSS
- 3) Although PSS cases for car-sharing and chemical management services have been investigated in depth, PSS applications to consumer products are few. Research into developing practical tools to help manufacturers realise sustainable PSS solutions is still at a formative stage.
- 4) In order to comply with the WEEE and RoHS directives, manufacturers/recyclers need to know comprehensive product life cycle information from all life cycle stages, especially at the usage and end-of-life.



### 2.3 Engineering Tools and Methods for PSS

This section reviews the literature related to engineering methods and tools for PSS, which cover technology domain of Life Cycle Engineering (LCE), service engineering, software engineering and Computer-Aided Design (CAD), etc.

#### 2.3.1 Life Cycle Engineering: Life Cycle Oriented PSS Design

Life Cycle Engineering (LCE) (Jeswiet, 2003) is usually used in the context of physical product design. LCE is “design and manufacture of products with the goal of protecting the environment and conserving resources, while encouraging economic progress, and at the same time optimising the product life cycle” (Jeswiet, 2003). Using LCE for the design of services resulted in the emergence of life cycle oriented PSS design (Aurich et al 2004), which includes life cycle oriented service design and PSS design.

- Life Cycle Oriented Service Design

Aurich et al (2004) claim that the design of services should be carried out in a simultaneous and integrated way with the physical design of product and manufacturing process. Product life-cycle costs, quality and performance are largely determined by product engineering. This simultaneous approach in engineering can reduce development costs, time and improve quality by taking advantage of mutual influence of products and production process. In order to facilitate the design of PSS, the focus of current life cycle oriented product and process engineering approaches need to be extended to include product, production process and design of relevant services in an integrated way.



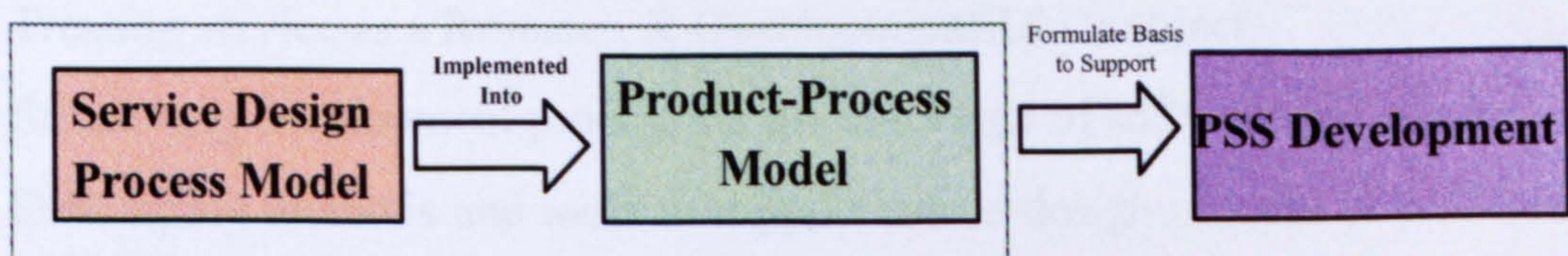
Figure 2.1 Service design process model (Wheelwright and Clark, 1992; Aurich et al, 2004).



A life cycle oriented reference process model for service design has been developed by employing the methodology of integrated design (Aurich et al, 2003), as shown in Figure 2.1. The process for the design of services is based on physical product design practice that is well developed even in a small manufacturing company. The following service design stages are therefore recommended (Wheelwright and Clark, 1992; Aurich et al, 2004).

- Customer demands identification
  - Technical and economic feasibility analysis
  - Concept analysis
  - Service modelling
  - Realisation planning
  - Prototypical service testing
- Life Cycle Oriented PSS Design

The life cycle oriented reference process model for service design can be implemented into any existing product-process model to formulate a foundation for establishing a life cycle oriented PSS design process (Aurich et al, 2004). For example, at the Institute of Manufacturing Engineering and Production Management at University of Kaiserslauten, Germany, an Integrated Product-Process Model (IPPM) was selected to incorporate the service design process model to support the design of PSS. This process is illustrated in Figure 2.2.



*Figure 2.2 Life cycle oriented PSS design (Aurich et al, 2004)*

The PSS design process model was developed by integrating the service design, product design and production design together. Although the approach is called the life cycle oriented PSS design, the product life cycle such as distribution, usage and end-of-life disposal, etc has not been taken into account. Main issues of PSS involving product usage and end-of-life treatment are not reflected. Furthermore, this approach does not



tackle the problem of how PSS can practically deliver services during a product life cycle.

### 2.3.2 Service Engineering: Product-Service Co-Design and Service Modelling

The term “*service engineering*” was first proposed in the mid 90’s in Germany and Israel (Bullinger, 2003; Mandelbaum, 1998). Service engineering can be understood as technical discipline concerned with the systematic development and design of services using suitable models, methods and tools. Service engineering promotes intensification of service content within a product life cycle. It emphasises dematerialisation by adopting a better technical-methodological approach and a more efficient use of existing engineering know-how (Tomiyama, 2001; Bullinger et al, 2003). Service engineering looks at development, design and production of services and aims at improving the efficiency and quality of the service as well as its delivery. Product-service co-design and service modelling are two approaches of service engineering.

- Product-Service Co-Design

Product-service co-design states that traditional engineering methods and tools in applied science can be borrowed for service design and development (Ganz et al, 2004). Main points of the product-service co-design are as listed as follows:

- Treating service as a Research & Development(R&D) object
- Structuring development process for the co-design of software and services
- Developing methods and tools to support the co-design process of software and services

Product-service co-design has many similarities to life cycle oriented design for PSS discussed previously. They both promote using systematic way to design and develop product and service by employing existing engineering models, tools and methods (e.g. water fall model). Actually these two approaches overlap somewhat. This approach has

same problems as life cycle oriented PSS design, e.g. how PSS can practically deliver services during a product life cycle is not addressed.

- **Service Modelling**

In the article “*Service Engineering to Intensify Service Contents in Products Life Cycles*”, Tomiyama claims that service modelling will serve as a basis for establishing service engineering to design, produce and develop services (Tomiyama, 2001). Although service modelling is yet a research issue, some core elements for modelling service have been identified as follows:

- *Service goal*. The goal of service is to satisfy the needs of the service receiver by changing its state.
- *Service Environment*. Service takes place within an environment that contains service provider, service receiver, and service channel in which the service content is sent.
- *Service Channel*. The service channel influences the quality of service (e.g. timeliness, frequency, punctuality, convenience)

Although core elements for the service modelling have been defined, modelling a service is still a tough task as there is less supporting tools to facilitate the service modelling. Implementation of modelling tools requires knowledge about activities, service environment, and service channel (Tomiyama, 2001).

### **2.3.3 Software Engineering: Scenarios and Use Cases**

The methods of “*use cases and scenarios*” in software engineering can be adopted in the design and development of PSS (Morelli, 2002). Use cases and scenarios are useful methods in software engineering to help design and develop the architecture of information management system (Bennett et al, 2002; Lee et al, 2002; Schneider et al,



1998). Use cases describe the visible requirements of a system and contribute to test plans and customer guides (Schneider et al, 1998). It defines a set of interactions between external actors and the system (Schneider et al, 1998). An actor is a person, organisation, or external system that plays a role in one or more interactions with the system. A particular scenario is a specific example of a use case (Douglass, 1999). The advantage of such scenarios is that domain experts and developers can easily understand the analysis through many typical system usage scenarios (Bennett et al, 2002; Lee et al, 2002; Schneider et al, 1998).

According to Morelli (2004), the development of PSS scenarios is based on the profile of the actors and the possibility that different actors may involve different configuration of the PSS. Each scenario consists of a number of use cases that describe the sequence of actions for each function within the scenario. Use case diagram can be used to describe the flow of events, relevant actors, pre- and post-conditions of each use case and alternative paths. Once scenarios and use cases are defined, PSS need to be further explored to understand the functions. The technique of Integration Definition for Function Modelling (IDEF) was suggested to model activities, actions, processes and operations required by PSS, the functional relationships and data that supports the integration of the functions.

The benefits of use case and scenario used in the design of PSS are as follows:

- A simple and easy start point at the PSS design stage.
- Functionalities provided by PSS are straightforward.
- How various stakeholders interact with PSS can be identified.

Negative points are summarised as follows:

- Appropriate for PSS function and operation design at definition stage, but not preferable at implementation stage
- Not appropriate for realising PSS
- Not address the issue of how services are created

### 2.3.4 Service CAD

CAD stands for Computer-Aided Design (CAD). Service CAD is a computer-based tool that can be used to design the service and PSS, just as CAD can be used to facilitate the design of products and simulation of their behaviours under various circumstances (Tomiyaama, 2003). Tomiyama has proposed a scheme to model services, which was discussed in the previous section (i.e. service modelling). Within the scheme, service is an action that changes the state of the service receiver. Service CAD can support both managers and engineers to model, design and develop services as well as PSS.

Tomiyaama claims that service CAD can be developed based on the modelling scheme that includes a service provider, a service receiver and a service channel. The development process of service CAD is summarised as follows:

- Theoretical development
- Actual development of a service CAD system
- Collection of actual service design cases and verification through test cases
- Actual large scale deployment, particularly in industry

It is a good idea to adopt CAD technique into the design of service and PSS. The bottleneck is that currently there is no service CAD software package available. Although the development process of CAD tool has been identified, developing a moderately appropriate service CAD software tool is still challenging. It requires developers to simultaneously understand the knowledge of service engineering, service modelling and software engineering.

### 2.3.5 New Service Development

The driver of the emergence of New Service Development (NSD) (Fitzsimmons et al, 2000) is that the product development paradigm fails to address the unique characteristics inherent in services, such as customer participation in the service process, intangibility, and heterogeneity of customer demand. The new service is defined as “an offering not previously available to customers that results from the addition of offerings,



radical changes in the service delivery process, or incremental improvements to existing service packages or delivery processes that customer perceive as being new” (Fitzsimmons et al, 2000).

The understanding of NSD is driven by the understanding of NPD (New Product Development). On one hand, in goods-based industries NPD has been widely studied, and the science of NPD continues to be examined and refined (Wind & Mahajan, 1997), on the other hand, there is little understanding of how new services are or how new services should be developed (Bowers, 1989; Edgett, 1994). Thus a NSD process cycle was introduced to address the need, as shown in Figure 2.3.

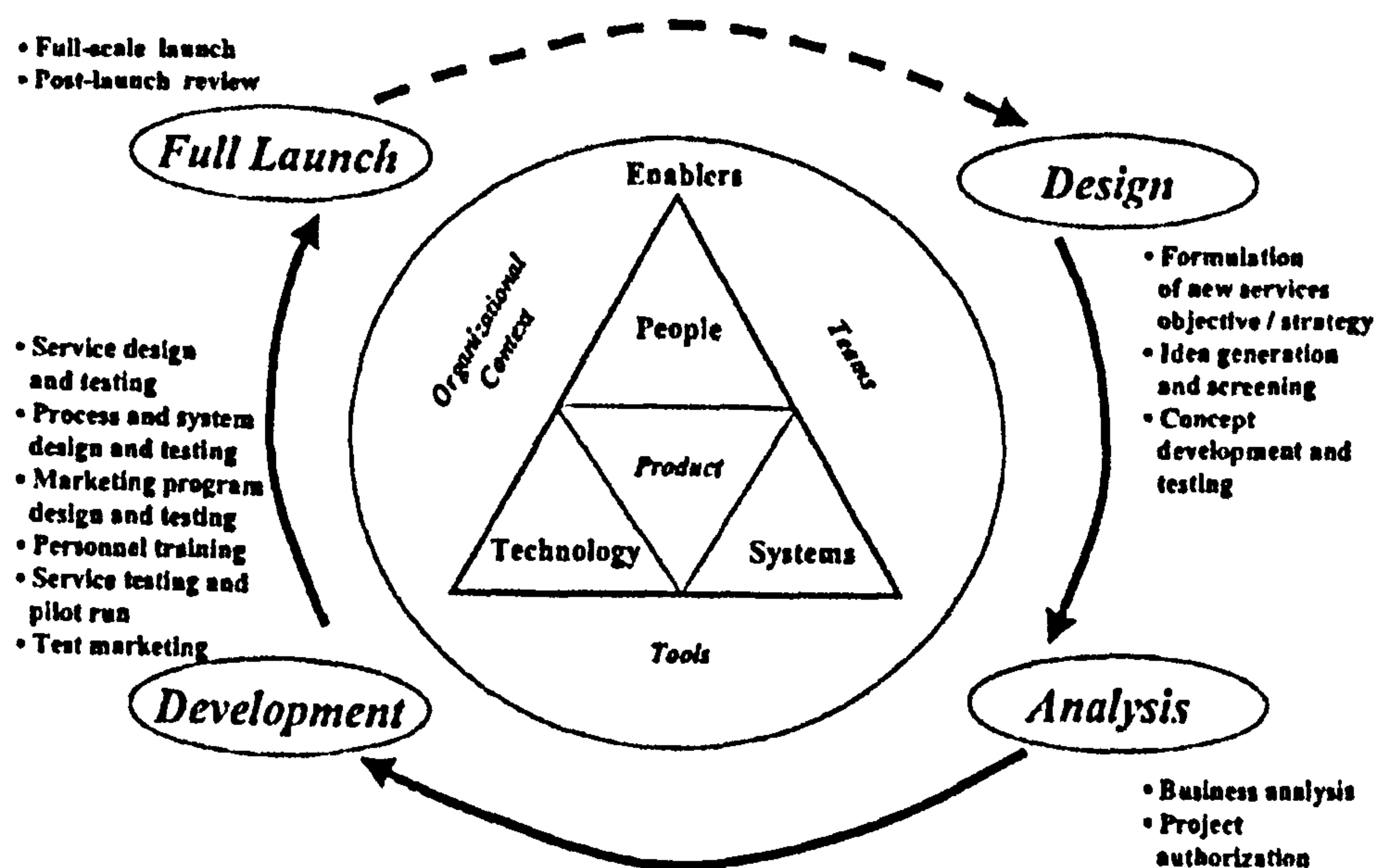


Figure 2.3 The NSD process cycle (Fitzsimmons et al, 2000)

The NSD process cycle takes into account some of the unique features of services (e.g. customer participation, intangibility) and demonstrates the importance of matching service delivery system design along with development activities. It also illustrates the importance of “enablers” (e.g. teams, design tools, organisation culture) to facilitate the design or redesign of the service delivery system.

### 2.3.6 MEPSS

MEPSS (MEthodology for Product Service Systems) is an EC funded project which aims to provide industry with a toolkit that enables it to analyse newly developed product-services with regard to design and implementation aspects, economical impacts, social and environmental impacts and issues related to consumer acceptance and culture and ethics (Pi-Net, 2002). It is one of the participating projects in Pi-NET (Product service system information NETwork) project.

The Pi-NET was initiated to facilitate the co-operation between 6 projects that were funded under the “Competitive and Sustainable Growth” program of the EC. These clustered projects are briefly introduced as follows (Pi-Net, 2002):

- ELIMA: *Environmental Life cycle Information Management and Acquisition for consumer product*, which aims to develop ways of better managing the life cycles of consumer products (ELIMA, 2004). More details of ELIMA project can be found in a later section.
- IN-TIME: *INtegrated storage and e-commercialisation system for the fast production and delivery of TIMber components*. The project will introduce methods of storage which is accessible even to small operators. These methods and machines will be fully automatic thus providing the company with an immediate visibility of all its products.
- MIRTH: *Musculo-skeletal Injury Reduction Tools for Health and safety*. The project aims to develop new and improve existing Musculo-skeletal Injury Reduction Tools for Health and safety.
- NEREFITE: *New Fire Retardant Textiles*. The project aims to enhance halogen free Fire Retardant (FR) properties for textile materials and develop intumescence halogen free FR to face evolution of textile European fire regulations.



- VIP: *Development of super Vacuum Insulating Panels and product integration service*. The proposed project suggests a new concept for thermal insulation of refrigerators, transportation boxes, cold storage rooms, buildings etc. The idea is to keep the insulating panels in vacuum inside sealed laminated bags (Vacuum Insulating Panels - VIPs).

The main objectives of the collaboration in Pi-NET project are to:

- Enhance the PSS-related aspects in *ELIMA*, *In-Time*, *Mirth*, *Nerefite*, *VIP* and *MEPSS*
- Improve the PSS methodology and tools to be developed in MEPSS project.

The intention of MEPSS is to provide an integrated theory through the joint effort of the EU consortium of respected experts in different disciplines. It will thus provide a business model and tools for industry, and may serve as a methodological framework to foster future research and support the implementation of PSS across European business sectors (Pi-Net, 2002). The focus of MEPSS toolkit is primarily on the analysis of newly developed product-services.

### 2.3.7 Summary

The present engineering tools and methods for PSS are mainly concerned with the general design and development of service and PSS. Broadly speaking, engineering methods of *service CAD* and *NSD* also belong to the discipline of service engineering besides *product service co-design* and *service modelling*. They highlight design and development of services. *Life cycle oriented PSS design* and “*scenario and use case*” are focused on the general design and development of PSS. *MEPSS* project is carried out to provide methods and tools to analyse the newly developed product-services. None of them has fully addressed the issue of how suitable services can be effectively and efficiently delivered during a product life cycle. Furthermore, the present engineering methods and tools can not help industry manufacturers to implement and realise sustainable PSS solutions.

## 2.4 Product Life Cycle Data and Acquisition

Product life cycle data is a valuable asset, which can provide knowledge of manufacture, transport, use, maintenance and disposal of the product. This can enable better decisions on resource use and re-use (Moore et al, 2000; Simon et al, 1998; Simon et al, 2000; Kiritsis, 2004; Simon et al, 2004). For example, Life Cycle Assessment (LCA) of the product requires accurate life cycle data to determine the energy and resources used. The systematic log of break-downs, repaired or replaced components, etc. can help maintenance providers to make decisions. The product usage data can help reuse/recyclers to predict the remaining life span of component(s) (Moore et al, 2000).

### 2.4.1 Product Life Cycle

Product life cycle can be classified as the following three stages: Beginning-of-Life (BOL), Middle-of-Life (MOL) and End-of-Life (EOL) (Kiritsis, 2004). Product life cycle data features different at each stage.

#### 2.4.1.1 Beginning-of-Life (BOL)

According to Kiritsis (2004), BOL phase includes design, component/material and manufacture. Life cycle data in BOL phase includes Bill of Materials (BoM), component/material properties, reliability test results and build inventory (i.e. the type and date component/parts installed) etc. The components/parts ID, the life of the components/parts and the use duration/times of components/parts can be used to make decisions about which components/parts are suitable for reuse / recycle at the end of the product life.

As mature technologies such as CAD/CAM are widely used at this stage, the data and information flow in BOL (i.e. design, component/material and manufacture) is fairly complete and easy to collect. Data in this stage rarely changes during the whole product life cycle (Kiritsis, 2004).

#### 2.4.1.2 Middle-of-Life (MOL)

MOL phase includes distribution, use and maintenance (Kiritsis, 2004).



- *Distribution*

During the distribution stage, product may suffer damage due to some unexpected reasons (e.g. mechanical shock). Thus information about events which occur due to unforeseen circumstances can be captured and added to a knowledge base. It can be used to identify and rectify the offending problem. In addition, preventive measures can be taken to avoid a reoccurrence of a similar situation (Moore et al, 2000; Simon et al, 1998; Simon et al, 2000; Simon et al, 2004).

- *Use*

The life cycle data during the use stage has huge potential to create values for stakeholders involved, especially manufacturers and recyclers (Moore et al, 2000; Simon et al, 1998; Simon et al, 2000; Simon et al, 2004). For example, appropriate analysis of life cycle data can lead to knowledge about

- Use pattern
- Product and component life span
- Advance warranty about remaining product /component life

Figure 2.4 shows an example of use pattern analysis. A use pattern of average door opening counts per day from 120 units of fridge/freezer was derived by analysing the correlation between the number of households and fridge/freezer door open times.

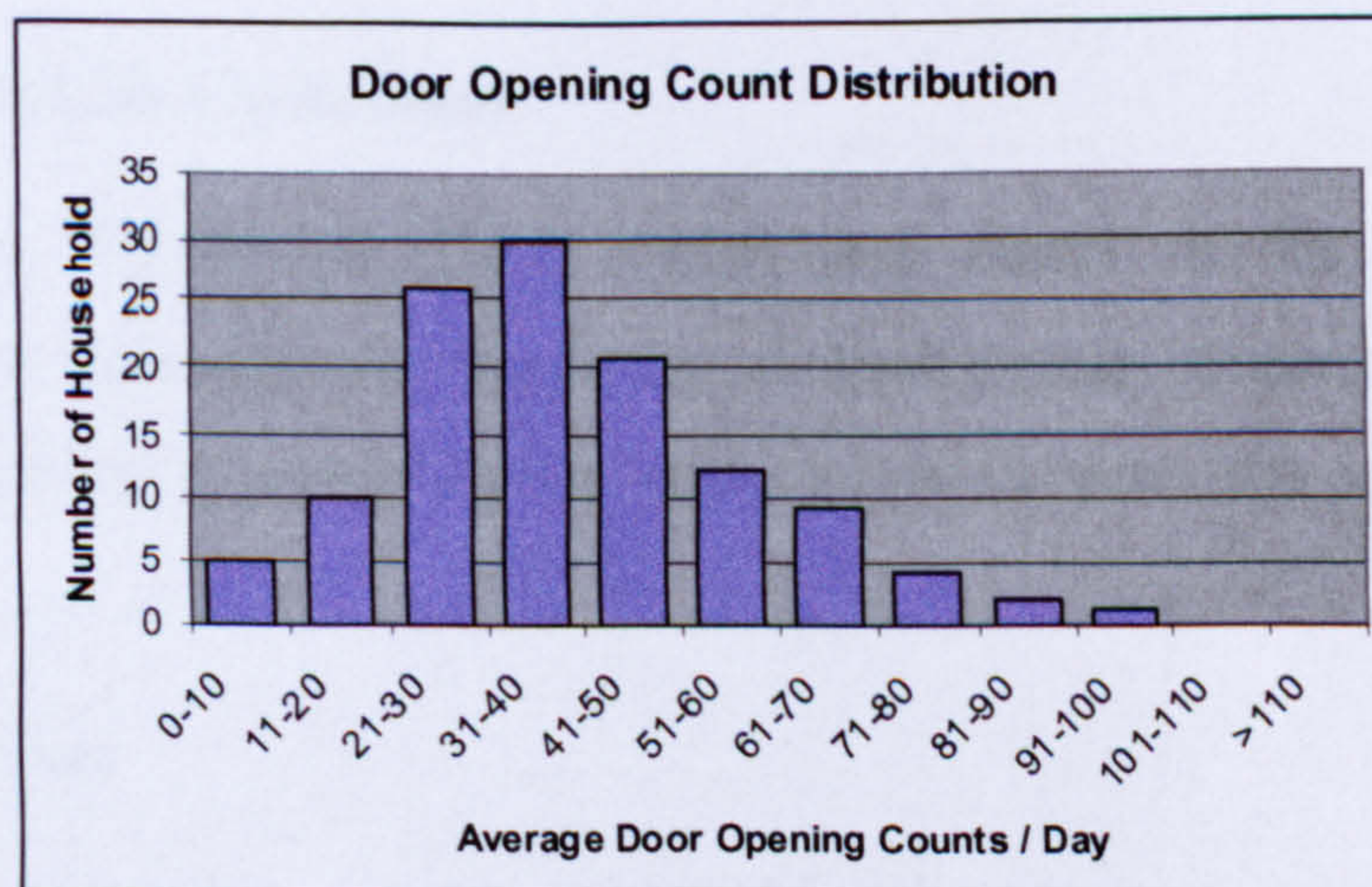


Figure 2.4 An example of use pattern analysis



- *Maintenance*

A product may require service / repair during its use-phase. Product life data can speed up fault identification, diagnosis and repair. Data related to repair / service may also support next generation product design. The collected life cycle data can be processed to help identify the most frequent problem areas to speed up troubleshooting and improve design.

#### 2.4.1.3 End-of-Life (EOL)

EOL phase includes refurbishment and reuse/recycle (Kiritsis, 2004). Three methods for reuse/recycle have been identified to be viable from the perspective of manufacturer (Cambridge Workshop, 2004; ELIMA Seminar, 2005):

- Product reuse: a product might be reused after repair and refurbishment.
- Component reuse: component might be reused for other products.
- Material recycles: materials may be recycled for new products.

At the end of product life, data collected at BOL and MOL stages has to be used to support the End-of-Life treatment, which is required by the new regulations and legislation in the EU, for example, WEEE and RoHS Directives described previously.

### 2.4.2 Product Life Cycle Data

Early work by Scheidt & Zong (1994) and others in the European CARE (Comprehensive Approach for Recycling of Electronics) consortium on electronics recycling defined two types of data that can be acquired: static data and dynamic data.

#### 2.4.2.1 Static Data

Static data, relating to the specification of the product, gives details of materials, components, configuration, options, instructions and suppliers. Static data is generated at the beginning of product life and rarely changes during the life time of the product



(Scheidt & Zong, 1994). Normally the static data includes information on: hazardous materials, material content, recoverable components, disassembly attributes (e.g. sequence and tools) and end-of-life disposal.

#### 2.4.2.2 Dynamic Data

The product-specific dynamic data is recorded during its life and may be continuously updated. Dynamic data recording could start as early as when the product is being transported. It is initially stored in the memory of the data acquisition device (Moore et al, 2000). Dynamic data is obtained from sensors and represents the measured parameters or values calculated from the measured data. Dynamic data can be also entered manually by the user, e.g. the timely information of initial registration, or service/maintenance activities.

In general, product life and performance are likely to be affected by transportation, the environment where it operates and its functional operation (Moore et al, 2000). Environmental conditions include temperature, humidity, vibration, shock, fall, acceleration, radiation, salt, dust, sand, etc; operational conditions like on/off cycles, operation time, etc have major impacts on the functionality and reliability of product (Mueller, 2003).

The decision on reusability and recycling may depend on the product status resulting from these environmental and operation conditions (Mueller, 2003). Knowing these conditions makes it easier to conclude the product status. The environmental/operating parameters mentioned above could result in oxidation, structural changes, softening, drying out viscosity, reduction, expansion, drift parameter, noise, insulating resistance, electrolysis, absorption, diffusion, leakage currents, ice formation, hardening, brittleness, increase in viscosity, contraction, fracture, loosening of connections, fatigue, and contact problems, etc. (Mueller, 2003).

## 2.4.3 Product Life Cycle Data Acquisition

### 2.4.3.1 Intelligent Data Unit (IDU)

An IDU (Moore et al, 2000; Simon et al, 1998; Simon et al, 2000) is an electronic device which is mainly responsible for data acquisition from consumer product. An IDU can be either an add-on device within the product (suitable for legacy product), or an embedded device. It is likely that the IDU capabilities may be built-in to the product when data acquisition and communication become essential requirements at early stage of product development. Normally the IDU requires the following components (Moore et al, 2000; Simon 1998):

- Transducers or sensors, to convert readings of environmental parameters such as temperature or vibration into a suitable form (analogue voltage or digital signal).
- A processor and clock, to receive the control signals, manipulate the data and store results
- Non-volatile memory
- A communication link to transfer the data

The idea of IDU that was used for the storage of lifetime data in order to facilitate the treatment or the reuse of the electronics devices emerged in 1994 (Scheidt et al, 1994). However, it was difficult to integrate such functions into electronic and electric products due to the high cost at that time. Other researches on IDU include using IDU on electric motors in power tools with a view to increase the motor reusability (Klausner et al, 1998a, 1998b) and recover resources from appliances (Grudzien, 2000). A group at Technical University of Berlin tested the concept of collecting data from fasteners to assist end-of-life disassembly (Seliger et al, 1999). The design of IDU could incorporate the automatic identification (Auto-ID) technology (e.g. Radio Frequency Identification). Auto ID is concerned with the automated retrieval of identity of objects and the information about the item to be stored, retrieved and communicated (McFarlane et al, 2003).



#### 2.4.3.2 “Whitebox” Project

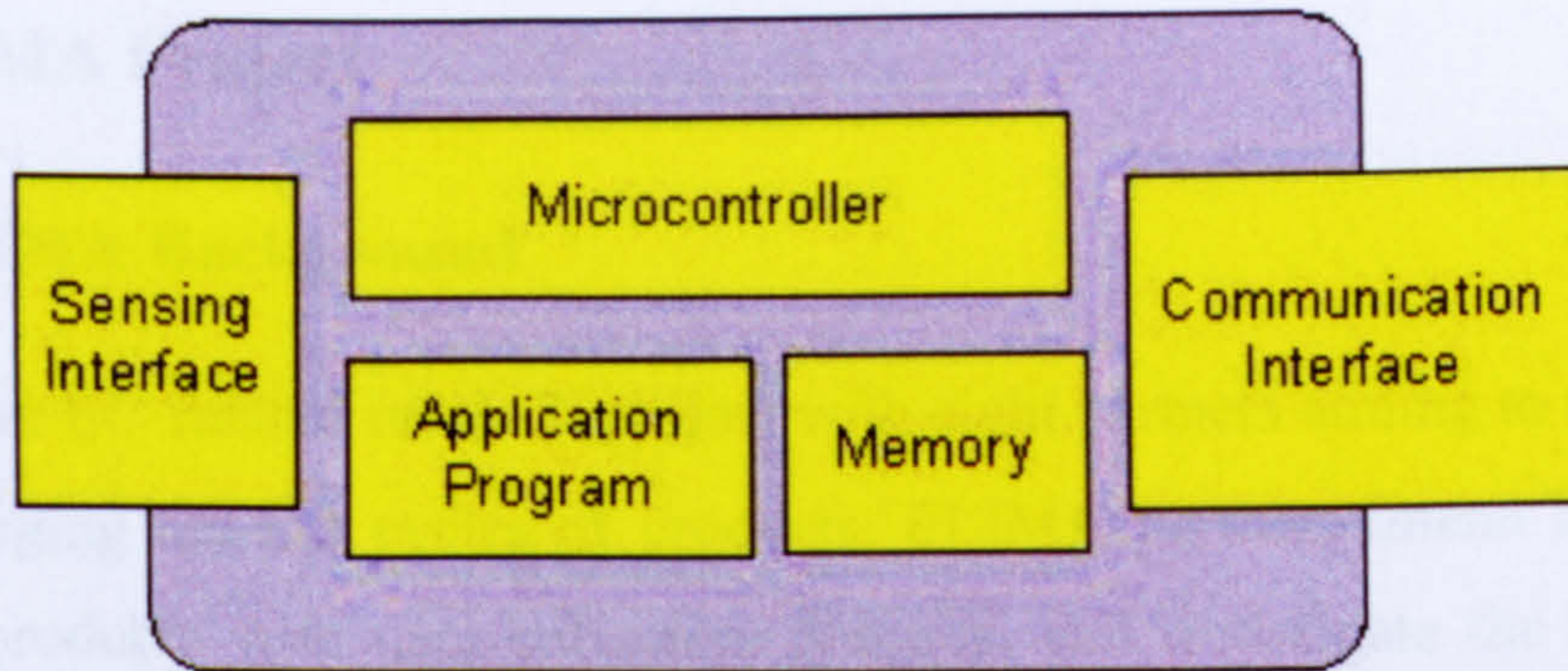
Requirements from European Commission through WEEE and RoHS Directives have been an incentive for a project called “Whitebox” by analogy with aircraft “black boxes”. An EPSRC funded project “*Life-cycle Data Acquisition and Devices for Consumer Products and Machines*” (i.e. Whitebox project) has been carried out to investigate the prototype IDU devices and assess the quality and value of the data obtained for product life cycle management (Moore et al, 2000; Simon et al, 1998; Simon et al, 2000).

Two types of IDU prototypes have been developed at De Montfort University and Sheffield Hallam University. At De Montfort University, the IDU prototype was developed to support the data acquisition in real time. The IDU prototype includes an interface board and a Lonworks control module, and power line to twist pair router which is used to transmit the acquired life cycle data to a computer (Moore et al, 2000). At Sheffield Hallam University, the prototype was built based on PIC17C756 microcontroller which is connected to a separate EEPROM (Simon et al, 2000).

The “Whitebox” project has demonstrated the significant environmental benefits of using the IDU devices, components or systems, which includes (Moore et al, 2000; Simon et al, 1998; Simon et al, 2000):

- Smart control of the appliance can reduce the resources consumption
- Preventive maintenance allows servicing and adjustment of product for optimum efficiency
- Removed parts and end-of-life recovered parts can probably be assessed for more effective reuse





*Figure 2.5 IDU reference architecture (Moore et al, 2000)*

Furthermore, in “Whitebox” project, a requirement specification of IDU has been identified as follows, shown in Figure 2.5:

- Data acquisition unit (normally a micro-controller ) and its functions
- Sensing requirements
- Memory requirements
- Interface of the data acquisition functions
- Interface of data readout /transmission

#### **2.4.4 Summary**

Although the “Whitebox” project has identified that the product life data can provide significant social, economic and environmental benefits, the infrastructure of how these benefits can be realised is not clear. The “Whitebox” project is mainly concerned with the life cycle data acquisition, i.e. how to acquire the product life cycle data, and the design and implementation of two IDU prototypes. An integrated view of the whole product life cycle to close the information loop in different phases of product life cycle is too weak. The lack of a closed information loop in the life time of a product is a fundamental problem. In general, information available at one stage (e.g. BOM at BOL stage) is lost and has to be regained at another stage (e.g. end-of-life). Industry needs systems that can integrate information to manage the product life cycle, to reduce costs and to improve the design of future consumer products and development of the new services.



## 2.5 ELIMA Project

### 2.5.1 ELIMA Background

ELIMA is an EC-funded research project with eight partners aiming to develop ways of better managing the life cycles of products. ELIMA partners intend to build and test consumer products with data collection features, and investigate the implications for manufacturers, users, and recyclers (ELIMA, 2004; Bodenhoefer et al, 2004). The technology includes sensors and memory built into the product to record the dynamic data such as time, temperature and power consumption. The acquired dynamic data together with static data will be transmitted over the Internet to a database where the manufacturer can actively manage the entire product life cycle. ELIMA's important innovation is the utilisation of an integrated information system to manage the entire life cycle. ELIMA is a particularly timely project due to the recent EU-wide adoption of the WEEE as well as RoHS directives and the current development of community Integrated Product Policy (IPP).

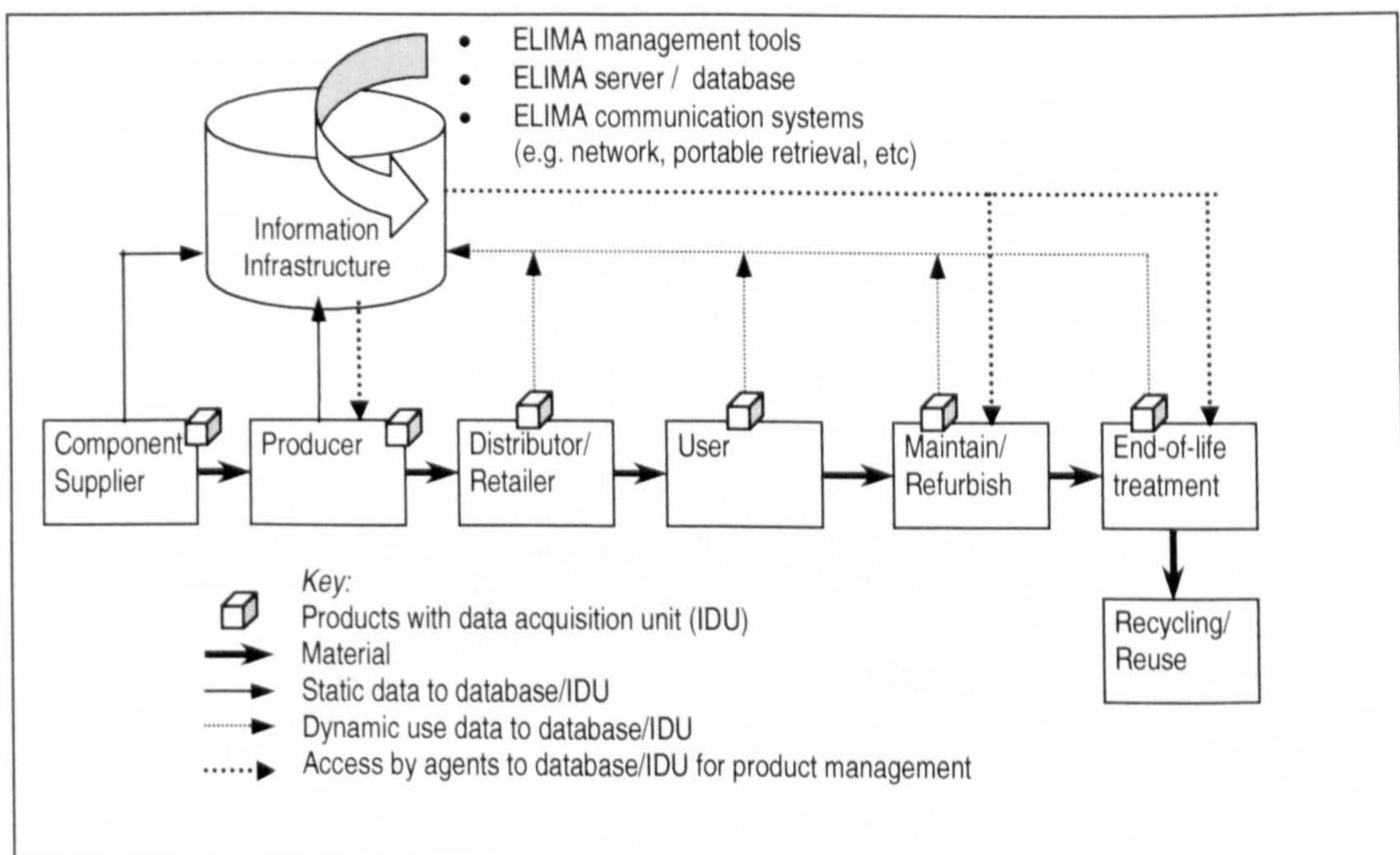


Figure 2.6 Product, data and information flow in ELIMA (ELIMA, 2004)

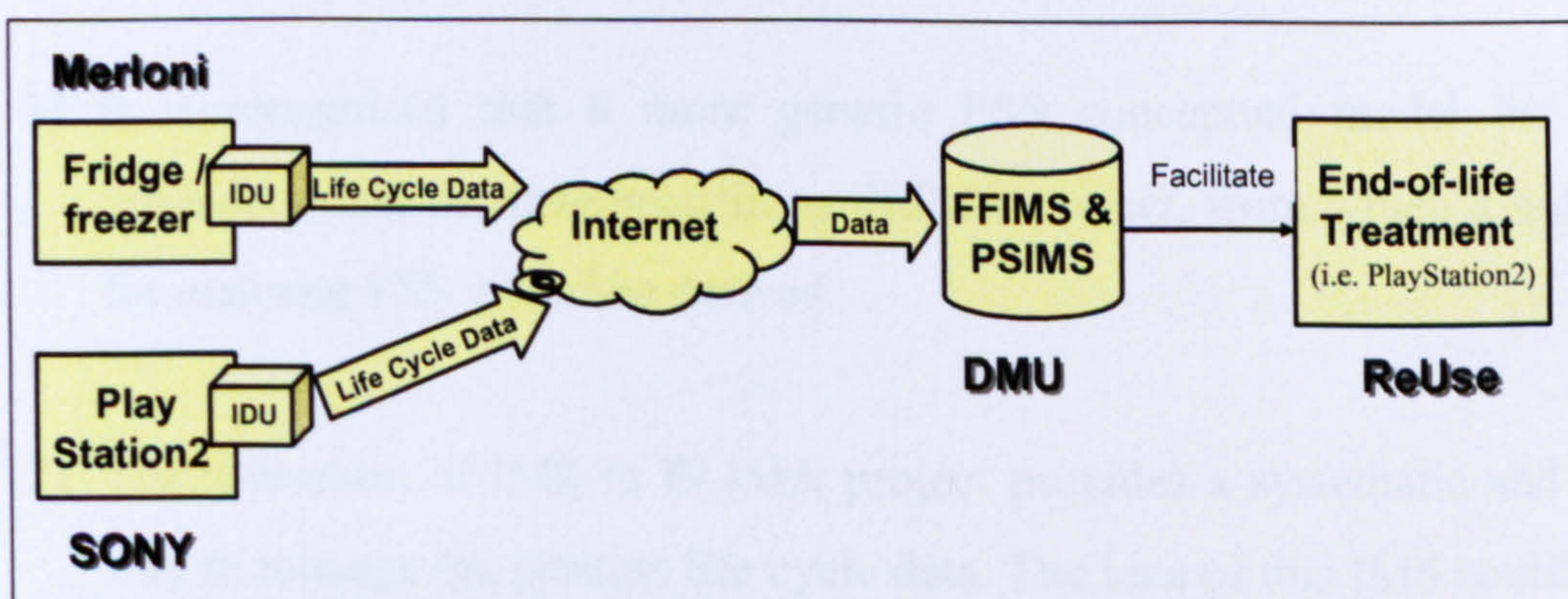


The consortium members include two manufacturers of consumer goods (i.e. Sony and Merloni UK), four organisations (i.e. Motorola, Cybernetix, ReUse, and SAT) supplying technology and know-how in communications, electronics and end-of-life treatment, and two universities (i.e. De Montfort University and Sheffield Hallem University).

The objectives of ELIMA include (ELIMA, 2004; Bodenhoefer et al, 2004):

- Performing two field trials, i.e. white goods field trial (i.e. fridge/freezer) and brown goods field trials (i.e. SONY Play Station2) to explore the solutions for industries to meet WEEE and RoHS Directives.
- Developing RFID smart tags-based IDU and GSM systems-based IDU for dynamic life cycle data acquisition and communication
- Developing an environmental life-cycle Information Management System (IMS) and associated management tools.
- Developing and validation of life cycle models for the assessment of cost and environmental benefits
- Evaluation of ELIMA concept and benefits

### 2.5.2 ELIMA Field Trials



*Figure 2.7 ELIMA field trials overview*

Figure 2.7 shows an overall structure of ELIMA field trials. In field trials, Merloni Elettrodomestici UK (Merloni UK) (<http://www.gda.uk.com>) uses fridge/freezer as an exemplar product of white goods and Sony International Europe (<http://www.sony->



[europe.com](http://europe.com)) uses electronic game console (PlayStation2) as an exemplar product of brown goods. In order to collect the life cycle data, two types of IDUs (i.e. EEPROM-based and GSM Modem-based) are to be developed and fitted into fridge/freezers of different product models; one type of IDU is to be developed for PlayStation2 (PS2) by employing Radio Frequency Identification (RFID) technology and some auxiliary sensory and interface components. De Montfort University (DMU) (<http://www.dmu.ac.uk>) is responsible for developing two IMSs, namely, Fridge/Freezer IMS (FFIMS) and PlayStation2 IMS (PSIMS) to house and manage the gathered life cycle data. The acquired life cycle data from fridge/freezer and PS2 is transmitted to FFIMS and PSIMS accordingly which will be running at the Intelligent Machines and Automation System (IMAS) laboratory at DMU. With the help of IMSs, ReUse (<http://www.re-use.net>) utilises the prototype treatment facility to handle end-of-life product treatment (e.g. identify hazardous materials, predict remaining lifetime).

### **2.5.3 Research Study in Association with ELIMA**

ELIMA is one of the Pi-NET cluster projects. There is a potential in the ELIMA project: using an integrated information system to manage the entire product life cycle can facilitate the product related services delivery. Hence this PhD research study is carried out in association with the ELIMA project, due to the following reasons:

- 1) It is recognised that a more generic PSS conceptual model for consumer products could be developed from ELIMA project, from which a methodology for realising PSS would be derived.
- 2) The utilisation of IMS in ELIMA project provides a systematic and integrated way to manage the product life cycle data. The idea of this IMS could be further extended and refined to become a construct of PSS for consumer products.
- 3) As ELIMA project requires FFIMS and PSIMS, both of which have many similarities, to be developed, generic software tools which can facilitate the



development of these two IMSs would then be developed. These generic tools could be further extended and refined as toolkit for realising PSS.

- 4) Facilities such as equipment (e.g. fridge/freezer, PS2, IDU), computer hardware and software (e.g. high-performance server, Macromedia Dreamweaver MX), communication channel and partner expertises that are employed in ELIMA project can be used to support this research study.
- 5) ELIMA field trials can be used as a test bed to provide an initial opportunity to evaluate the proposed methodology and associated toolkit.

## **2.6 Chapter Summary**

- 1) The present engineering methods and tools for PSS are mainly concerned with the general design and development of service and PSS. They have not yet fully resolved the issue of how services can be delivered during a product life cycle. In PSS research, there has been very little work carried out to develop systematic and integrated engineering tools which can help manufacturers implement and realise PSS solutions in a practical way.
- 2) It has been identified that PSS can facilitate the realisation of sustainability through the following two ways: (i) provision of a product service mix rather than just a physical product because service can create added value with less environmental impact; (ii) advocating a new business model - “to have a utility instead of an ownership”.
- 3) The oncoming enforcement of European Union WEEE and RoHS Directives require manufacturers to take the responsibility for the collection, disposal and recovery of the end-of-life product as well as to restrict use of hazardous substances. In order to comply with directives, manufacturers and recyclers need to know more product information at each life cycle stage.



- 4) The product life cycle data has been recognised as a valuable asset. Although life cycle data of consumer products can be acquired, the infrastructure to realise the value is not clear. Furthermore, an integrated view of the whole product life cycle to close the information loop in different phases of product life cycle is too weak.
- 5) The background of ELIMA project is introduced. The reason why this research study is conducted in association with ELIMA project was explained.



## Chapter 3 Literature Survey II - Enabling Technologies for Product Life Cycle Information Management

### 3.1 Introduction

In order to realise a service enabling scheme and PSS solution for consumer products, the following major areas have to be covered:

- Product life cycle database analysis and design
- Service enabler framework analysis, design and implementation
- Life cycle data presentation, communication and exchange
- Web-based service enabler

Therefore, in the context of this research, 5 topics of enabling Information and Communication Technologies (ICT) are broadly identified as follows:

- (i) Database and Information Management System
- (ii) Information System Development and Software Framework
- (iii) eXtensible Mark-up Language (XML) and Web Services
- (iv) Web Technologies
- (v) Automatic Identification (Auto-ID) Technologies

### 3.2 Database and Information Management System

The database approach now dominates nearly all of computer-based information systems used today (Jessup et al, 2003). Database and information management system are now such an integral part of our day-to-day life (e.g. purchases from the supermarket, purchases using your credit card, booking a holiday at the travel agents, using the library, using Internet) that often people take their use for granted (Jessup et al, 2003).



### 3.2.1 Database and Database Management System (DBMS)

The database technology underpins the fundamental advances in communication systems, transportation and logistics, financial management, knowledge-based systems, accessibility to scientific literature, and a host of other civilian and defence applications (Silberschatz et al, 1996). Sometimes the terms - database, database system, Database Management System (DBMS) are used interchangeably. According to Connolly (2002), the database is a shared collection of logically related data, and a description of this data, designed to meet the information needs of an organisation. The DBMS is a software system that enables users to define, create, maintain and control access to the database. Database system encompasses a database and a collection of software application programs that interact with database. The relationship between database and DBMS is shown in Figure 3.1.

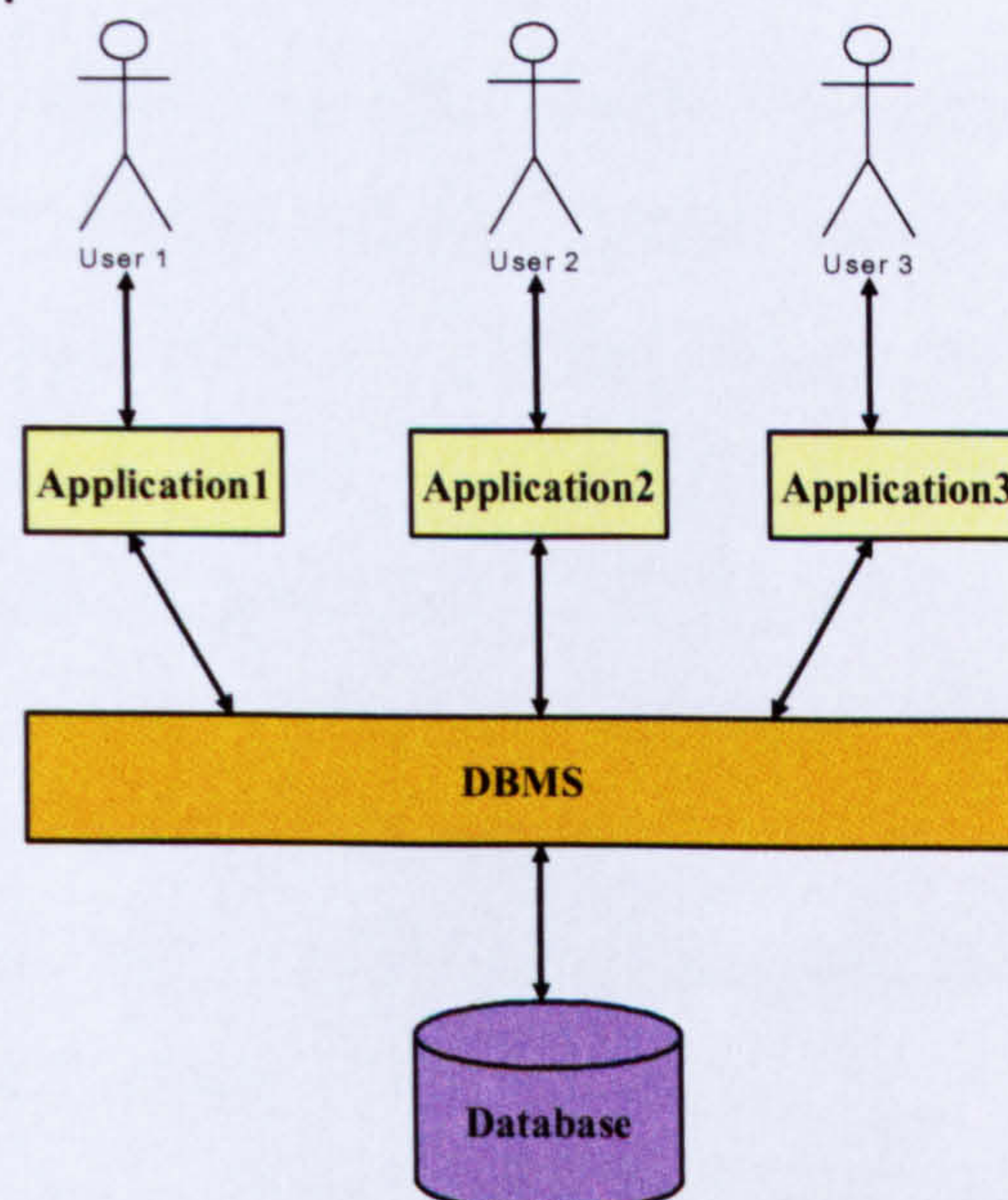


Figure 3.1 Database and DBMS (Rolland, 1998)

According to Rolland (1998), DBMS provides the facilities for:

- Data management
- The sharing and integration of data between different applications
- The ability to support multiple views of the same data
- Controlled concurrent access to data
- Ensuring the security and integrity of data



### 3.2.2 History of Database and DBMS

It is suggested the DBMS has its root in 1960s Apollo moon-landing project, where vast amount of data and information generated by the project needed to be handled and managed. As a result, North American Aviation (NAA, now Rockwell International) which was the main contractor for the project, developed software known as GUAM (Generalized Update Access Method). In mid 1960's, IBM joined NAA to develop GUAM into what is known as IMS<sup>1</sup> (Information Management System) (Connolly, 2002). This was an early attempt to achieve the aim of database approach to file management and is one of the earliest examples of DBMS. In IMS, a database is conceptually represented in hierarchy thus IMS is a typical example of hierarchical DBMS which is still being used by most large mainframe installations.

In 1970's, the Conference On Data Systems Languages (CODASYL) set up a Data Base Task Group (DBTG) to define standard specifications for an environment that would allow database creation and database manipulation. A draft report was issued in 1969 and the first definitive report completed in 1971. For standardisation, the DBTG specified three distinct languages (CODASYL, 1975):

- A schema Data Definition Language (DDL), which enables database administrator to define the schema
- A subschema DDL, which allows the application to define the components of database they needed
- A Data Manipulation Language (DML) to manipulate the data

Although the report was not formally adopted by the American National Standard Institute (ANSI), a lot of DBMSs were subsequently developed following the DBTG proposal. These systems are known as CODASYL or network DBMS. The CODASYL and hierarchical approaches represented the first-generation of DBMS. These two DBMS have some fundamental disadvantages:

- Complex programs have to be written to answer the simple query

---

<sup>1</sup> This IMS refers to the IBM product, which is different from the product lifecycle IMS used through the whole thesis.



- Minimal data independent
- No widely accepted theoretical foundation

In 1970 Dr. E.F. Codd of IBM Research Laboratory produced his influential paper on relational database model (Codd, 1970). His relational model for database was noticeably different from previously described models and by the 1980's it has become generally accepted as the most coherent and usable model for DBMS development (Rolland, 1998). In late 1970's, System R project at IBM's San José Research Laboratory in California developed a Structured Query Language (SQL), which has since become the standard language for relational DBMSs (Astrahan et al, 1976). Currently there are several hundred relational DBMSs for both mainframe and PC environment. For example, DB2 and SQL/DS from IBM, Oracle from Oracle Corporation, INGRES II from Computer Associates, SQL Server and Access from Microsoft, open source MySQL, etc.

Relational DBMSs are referred to as second-generation DBMSs. The major criticism of relational model comes from its limitation in modelling capabilities. In 1976, Chen presented the Entity-Relationship model, which is now a widely accepted technique for database analysis and design (Chen, 1976). Codd himself also presented an extended version of relational model called RM/T (Codd, 1979) and RM/V2 (Codd, 1990) to address the failings.

In response to the increasing complexity of database application, two new systems have emerged: the Object-Oriented DBMS (OODBMS) and the Object-Relational DBMS (ORDBMS). The main feature of these two DBMSs is that activities can become part of the definition of database object rather than being separately encoded in applications. This evolution represents the third-generation of DBMS (Connolly et al, 2002).

### **3.2.3 Database Design: E-R Modelling and Normalisation**

Database design is the process by which the requirements of a database are modelled prior to implementation (Rolland, 1998). According to Rolland (1998), the process of



designing a large database can be broken down into two basic stages: (i) capture the database user's requirements and represent these in the form of a conceptual model (ii) convert the conceptual model to a conceptual schema that can be implemented on a given DBMS. A conceptual model is logical representation of a database at a higher level than a conceptual schema that describes the lowest database representation made available to a database user.

There are several conceptual modelling techniques for the database concept modelling, which includes Entity-Relationship (E-R) modelling (Chen, 1976) which enables the conceptual model to be built using entities and relationships between them, *Functional Data Modelling* (Shipman, 1981) which uses both diagrams and notation to represent a database as a series of mathematical functions, and *Semantic Object Modelling* (Kroenke, 1988) which uses diagrams to represent a database as a series of interacting semantic objects. Among these modelling techniques, E-R modelling is the most widely used form of database modelling due to its relatively simple and highly applicable to business type scenarios.

The E-R model of a database has three fundamental components:

- **Entities:** These are items in the real world that are capable of a unique existence (e.g. *customer* entity, *account* entity).
- **Attributes:** These are things that describe an entity (e.g. name of the *customer* entity; balance of *account* entity). An important attribute is the key attribute that uniquely identifies the entity.
- **Relationships:** A relationship represents the interaction between entities. A relationship has a degree indicating the number of entities participating in the relationship. Each interaction has cardinality.

After the E-R model has been transformed into a relational database that consists of a set of tables, normalisation techniques has to be employed to ensure that the given set of tables is an efficient implementation of a relational database. Normalisation is a formal technique for analysing relations based on their primary key and functional dependencies (Codd, 1972). The technique involves a series of rules that can test



individual relations so that a database can be normalised to any degree. These rules include:

- An efficient relational database should be absence of redundancy
- An efficient relational database should have minimal use of null values
- An efficient relational database should prevent the loss of information

#### **2.5.1.4 Information Management System**

An information management system is a system of inputting, collating and organising data that should provide selective data and report to the management, to assist in monitoring and controlling the project organisation, resources, activities and results (Jessup & Valacich, 2003). It acquires data from one or many disparate sources, and provides access to all who have share in that information or right to access that information. An Information management system includes a set of interacting components – actors, procedures, and technologies- that together collect, process, store, and distribute information to support control, decision-making and management in organisations (Silberschatz et al, 1996). As Web technologies are increasingly developing nowadays, most information management systems are web based or Internet based.

### **3.3 Information System Development and Software Framework**

As this research study involves the information system development and software framework development, literature related to information system development approaches, Object-Oriented (OO) analysis and design, Unified Modelling Language (UML) and software framework are reviewed in this section.

#### **3.3.1 Information System Development Approaches**

According to Bennett et al (2002), information system development approaches include (i) traditional waterfall life cycle development, (ii) prototyping development, (iii)



iterative and incremental development, (iii) the unified software development process and (vi) agile development.

- Waterfall life cycle development

In waterfall life cycle model, the process is subdivided into the formal stages where the output of one stage forms the input to the next. These formal stages include feasibility study, requirement analysis, design, implementation, testing, operation and maintenance (Vidgen et al, 2002). Feasibility study is concerned with defining the scope and purpose of the system, considering a range of alternative solutions and investigating the impact of the system. Requirement analysis aims to understand user requirements. The key in this stage is “what” instead of “how “. Design is concerned with translating the requirements into a software system specification that can be turned into working system by system developers (implementation). Once the system passes the acceptance test, it is transferred to live operations, where bugs and further improvements may be identified.

The main limitations of the waterfall life cycle development have been identified as follows:

- Real development rare follows such a sequential life cycle. System stages may overlap and activities may have to be repeated
- Once the system architecture has been made, it is difficult to make some change to user requirements
- Expensive, time-consuming and inflexible

- Prototyping development

Prototyping development overcomes many of the potential misunderstanding and ambiguities that may exist in user requirements. Prototype is a system or partially completed system that is built quickly to explore some aspect of the system requirements (Bennett et al, 2002).

Although prototyping has some advantages such as easy identification of any misunderstanding between developer and client, missed requirements can be identified,



feasibility and usefulness can be tested, etc, the problems are also obvious (Bennett et al, 2002):

- The client may perceive the prototype as a part of the final system, and may expect the whole system to be delivered soon
- The prototype may divert attention from functional to only interface issues
- Prototyping requires significant involvement of end user

- Iterative and Incremental Development (IID)

In order to overcome problems within waterfall model, an approach of Iterative and Increment Development (IID) was suggested by Gilb (1988). Gilb introduced the terms “evolution“ and “evolutionary” to the process lexicon and stated that the successful large system starts from successful small system that grows incrementally. The establishment of each increment provides feedback to the development team and informs the subsequent development. The landmark in IID publication was Barry Boehm’s “*A Spiral Model of Software Development and Enhancement*” (1988). The idea of spiral model is the evolutionary development which can help manage risks. It promotes that the developers should only define and implement the highest priority features, and then get feedback from end users or customers.

IID can be regarded as an evolution of rapid prototype development. It emerged to overcome the problems caused by the water fall development. It has now become a common software development practice and led to the emergence of Unified Software Development Process.

- Unified Software Development Process (USDP)

The Unified Software Development Process (USDP) (Jacobson et al, 1999) reflects the current emphasis on iterative and incremental life cycles. It incorporates the Unified Modelling Language (UML) and comprises much good common practice on software development. As illustrated in Figure 3.2, USDP development cycle comprises four phases, namely, *inception*, *elaboration*, *construction* and *transition*. Each phase may contain one or more iterations. The iteration has five workflows, namely, *requirements*, *analysis*, *design*, *implementation*, and *test*.



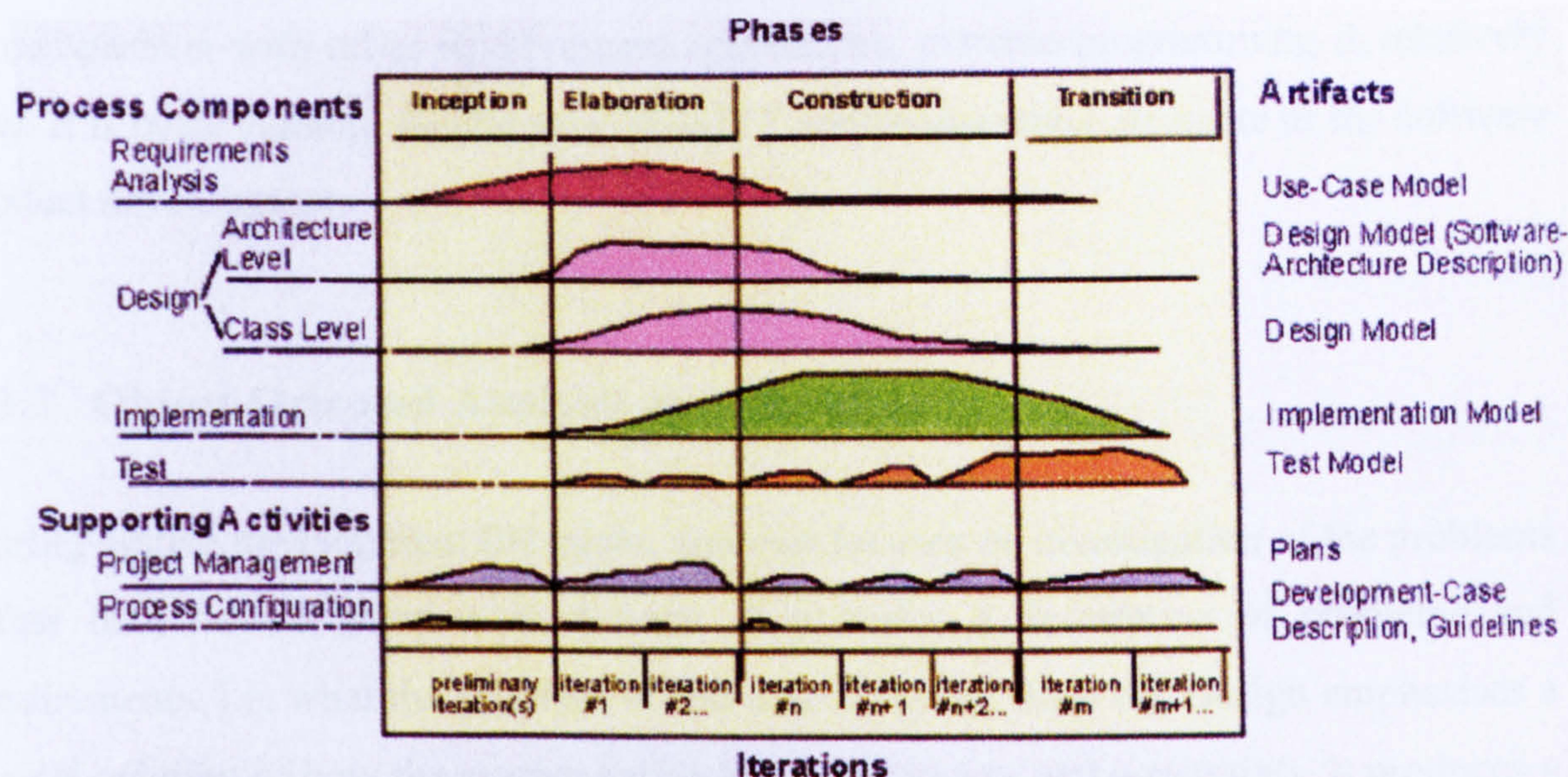


Figure 3.2 Unified software development process (Jacobson et al, 1999)

USDP is an industry standard software development process (Züehlke Engineering AG, 2002). It features that it is (i) use-case and risk driven, (ii) architecture centric, and (iii) iterative and incremental. The popular software development tool *Rational Unified Process* (RUP) created by Rational Software Corporation is an instantiation the commercialisation of USDP.

- Agile development

The main approach of agile software/system development is extreme programming (Beck, 1999). Extreme programming is a set of rules, principles and practice for rapidly developing high-quality software that provides the highest value for the customer. The approach is extreme in the sense that it takes 12 well-known software development best practice to their logical extremes. The premise of the extreme programming is the assumption that requirements change is inevitable. Rather than treat the changes to requirements as an exception, it is better to assume that requirements will change. The rules, principles and practice stipulated by extreme programming encouraging development teams to follow include pair programming, first delivery in weeks, feedback quickly, etc (Beck, 1999).



In comparison with other devolvment approaches, extreme programming is relatively new. It is more suitable for the specialised IT companies which dedicate to the software product development.

### 3.3.2 Object-Oriented Analysis and Design

During system development life cycle, analysis focuses on investigation of the problems rather than how a solution is defined. It provides a description of problems and requirements, i.e. what the problem is and what a system must do. Design emphasises a logical solution of how the system fulfils the requirements and constraints. It produces a high level and detailed specification of the logical solution to meet the requirements and constraints (Larman, 1998).

The traditional analysis and design approach is *structured analysis and design*, where the problem was de-composed by function or process to form a hierarchical breakdown of process composed of sub-process. The big problem of the structured analysis and design is that it makes software system hard to maintain because the structure of the software is based on its functionality, and fundamental change to what the system does will affect its underlying structure.

Object-Oriented (OO) analysis and design emphasises considering a problem domain and logical solution from the perspective of objects (e.g. things, concepts, or entities) (Larman, 1998; Britton et al, 2000). The logical software objects will eventually be implemented using an OO programming language (i.e. java, C++). OO analysis and design is now the principal industry-proven method for developing reliable, modular, testable programs and systems. Consistent use of OO techniques leads to shorter development life cycles, increased productivity and reduced system maintenance costs (Britton et al, 2000; Lee et al, 2002).

According to Lee et al (2002), 13 principles of OO approach are as follows:



- *Encapsulation.* An object contains both data and methods.
- *Information hiding.* An object defines how other objects access to its methods and data.
- *Message passing.* An object communicates with another object only through the message-passing mechanism.
- *Late binding.* The specific receiver of any given message is not known until runtime, also known as dynamic binding.
- *Delegation.* Client (one object) passes its work to agent (another object) via message passing. Agent provides the services that clients need.
- *Class and objects.* All objects are instances of a class. All objects of the same class use the same method in response to a specific request.
- *Generalisation/specialisation without polymorphism.* The specialised class (subclass) inherits attributes, relationships and methods from the generalised class (super-class).
- *Polymorphism.* The specialised class (subclass) inherits attributes, relationships and methods from the generalised class (super-class). However, a subclass may create its own method to replace the method with the same name in the super-class. The method of subclass overrides the super-class method for providing the same service.
- *Relationships.* Association and aggregation are used to capture the collaboration between objects necessary to provide service to a client.
- *Interface and objects.* All objects that implement an interface are also instances of that interface.
- *Generalisation/specialisation of Interfaces.* Specialised interface inherits the service protocol from the generalised interfaces.
- *Reflection.* Each object knows the detail information about the classes and interfaces.
- *Multithreading.* Each object can have concurrent execution paths. An object can handle multiple service requests in a concurrent manner.



### 3.3.3 Unified Modelling Language (UML)

Unified Modelling Language (UML) is a visual modelling language that is used to specify, visualise, construct and document the architecture of a software system (Bennett et al, 2002). It represents an important branch of OO approach. UML emerged from the unification that occurred in 1990s following various software development methodologies between 1980s and 1990s. Among these methodologies, the most significant ones are, Responsibility-Driven Design (RDD) (Wirfs-Brock et al, 1990), Booch method (Booch, 1994), objectory (Jacobson et al, 1992), and Object Modelling Techniques (OMT) (Rumbaugh et al, 1991). UML is considered as a third-generation OO modelling language. Although UML evolved primarily from various second-generation OO methods (at the notation level), its scope extended its usability far beyond its predecessors. UML is fast becoming an industry standard, with OMG (Object Management Group) acceptance, and a rich set of resources and software development tools available (e.g. Rational Rose modelling software).

UML provides a number of methods, tools and notation to describe and model the system, which includes use case, class diagrams, sequence diagrams, activity diagrams and state transition diagrams, etc described as below (Bennett et al, 2002; Lee et al, 2002; Schneider et al, 1998; Douglass, 1999):

- *Use case.* Use case is used to specify the functionality that the system will offer from the user perspective, and used to document the scope of the system. Use case diagram is one diagramming technique used to summarise the user's functional requirements in a high level overview.
- *Class diagram.* A class diagram is a model of the things that are of interests in the problem domain being studied.
- *Sequence diagram.* The sequence diagram models the collaboration between objects and actors, showing the exchange of message needed to accomplish a specific purpose. Typically, a sequence diagram is prepared to present a single use case.
- *Activity diagram.* Activity diagram can be used to model different aspects of a system. It can be used to model a task in business modelling, to model a system



function represent by a use case; to model a detail of how a specific operation is carried out, etc. Activity diagram is most useful to model business activities in the early stage of the project. Though sequence diagram is more close to the spirit of OO approach, there are some occasions that the activities must be carried out, but has not yet identified objects or classes involved to them. In such circumstances, activity diagram may be an appropriate tool to use.

### **3.3.4 Software Framework**

Software framework is a way of delivering system/application design pattern to support the best practice sharing during system/application development (Short, 1997). In a more general term, a software framework is a set of prefabricated software building blocks that programmers can use, extend, or customise for specific computing solutions. It proposes an architecture that is optimised for all applications within a certain domain and is designed to allow rapid instantiation of application within that domain (Gamma et al, 1995; Fayad et al, 1995).

The key characteristic of the framework is that it is designed to be customised prior to use. Besides providing glue logic in an application which is making use of the framework, the application designers are required to provide code called by the framework at critical points, known as plug points. This provided code “fills in the blanks” left deliberately by the framework’s designers. The customisation of the framework is through two main mechanisms described as follows (Short, 1997; Moore et al, 2002):

- A framework supplied components may be substituted by an application specific one.
- Using template file which provides a pre-defined design pattern to guide users to customise the framework by supplying parameters to the generic terms defined in the framework

Good examples of the software framework include Real Time Operating Systems (RTOS) which represent a highly successfully example of software framework for real-time applications (Mooney et al, 2002), and AOCS framework which applies to the



Attitude and Orbit Control Subsystem (AOCS) which is being used at Alenia Spazio to develop the software for the Antenna Pointing Control (APC) subsystem of the geostationary platforms (Pasetti et al, 2001).

### 3.4 XML and Web Services

Over the past few years, XML has enabled heterogeneous computing environments to share information over the Web (Vawter, 2001). It now offers a simplified mechanism to deliver the process as well, i.e. Web service. Web service is a natural evolution of XML application from structured representation of information to structured representation of inter-application messaging.

#### 3.4.1 XML: Extensible Markup Language

When Tim Berner's Lee created his original Internet hypertext application in 1990, he defined a fixed set of tags to do a fairly limited job. The restriction led to the development of XML in 1996 by World Wide Web Consortium (W3C) (W3C, 2000b). Now XML is already becoming the *de facto* standard for data communication within the software industry. XML is a restricted version of SGML (Structured Generalised Markup Language), designed especially for Web documents. It is a meta-language that enables designers to create their own customised tags to provide functionality not available with HTML.

XML is a standard for describing how data and information is structured (Connolly et al, 2002). Data and information presented in XML document is simple to read and understand, and it can be processed easily by computers. This makes it much more straightforward to move structured information from place to place or from one program to another. XML is also a standard language for interfacing with data. It performs a very similar function to ASCII (a standard way of encoding characters as bytes) or SQL (a standard way of querying to database). XML provides significant advantages over traditional approaches for inter-application message, interface



development and integration as follows: (i) separates content from presentation (ii) readability (iii) extensibility (vi) widespread industry adoption (v) knowledge and skills base (Connolly et al, 2002; Norris et al, 2001; W3C, 2000b).

XML itself is surprisingly simple, but a number of XML related technologies are rather complex (Norris et al, 2001). These technologies include Document Object Model (DOM) and Simple API for XML (SAX), the eXtensible Stylesheet Language (XSL), the eXtensible Stylesheet Language Transformation (XSLT), the XML Path Language (XPath), XML Schemas and Document Type Definition (DTD), etc.

### **3.4.2 Web Services**

According to Glass (2000), Web service is defined as “a collection of functions that are packaged as a single entity and published to the network for use by other programs”. Web services are building blocks for creating open distributed systems, and allow companies and individuals to quickly and cheaply make their digital asset available worldwide. Prior to the advent of Web services, enterprise application was very difficult due to differences in programming languages and middleware used within organisations. Interoperability was cumbersome and painful. With Web services, any application can be integrated as long as it is Internet-enabled. Web services provide a standard means of communication among different software application, running on a variety of platforms and/or/frameworks to ensure the interoperability (Connolly et al, 2002; Glass, 2000).

The foundation of Web service is XML messaging over standard web protocols such as HTTP. Web Services includes a suite of supporting protocols, namely, SOAP (Simple Object Access Protocol) transportation protocol, WSDL (Web Services Description Language) description language, and UDDI (Universal Description, Discovery and Integration) which is a Web-based distributed directory that enables business to list themselves on the Internet and discover each other(Connolly et al, 2002; Glass,2000).



## 3.5 Web Technologies

The World Wide Web (Web for short) has become the most popular and powerful networked information system today. As the Web has been designed to be platform-independent, many companies and organisations are now rapidly building Web-enabled information system as a strategic platform for implementing novel business solutions (Connolly et al, 2002).

Currently, various Web technologies, tools and platforms are available to support the Web development, such as *WebSphere*, *WebLogic*, *JBuilder*, *DreamWeaver*, *ASP*, *PHP*, etc. These Web technologies, tools and platforms can be generally categorised into three types: Java/J2EE based Web solution, Microsoft Web Solution and other Web solution (e.g. PHP, Oracle Internet Platform).

### 3.5.1 Java /J2EE

- Java

Java was developed by Sun Microsystems and originally intended as programming language appropriate for support an environment of networked machines and embedded systems. Java did not fulfil its potential until the Internet and the Web become popular. Now Java is rapidly becoming the *de facto* standard programming language for Web development.

Java is type-safe, object-oriented language that is interesting because of its potential for building Web applications (*applets*) and server applications (*servlets*) (Gosling et al, 2000). Java is practically interesting because of its Java Virtual Machine (JVM) (Arnold et al, 2000). The source code (.java) is compiled into byte codes (.class), which can be interpreted and executed directly on any platform to which JVM have been ported. The class file must be loaded from the local disk or from a network into memory by Class Loader before executed.



- J2EE

On one hand, J2EE is a specification, which defines a standard for developing, deploying and running enterprise applications and on the other hand, J2EE is a platform. The J2EE platform uses a distributed multi-tiered application model for enterprise applications. The J2EE application is divided into components which include: (i) client-tier components running on the client machine, (ii) Web-tier components running on the J2EE server, (iii) business-tier components running on the J2EE server and (iv) Enterprise Information System (EIS) tier running on the EIS server (J2EE,2004), as shown in Figure 3.3. J2EE multi-tiered applications are generally considered to be three-tiered applications because they are distributed over three locations, i.e. client machine, J2EE server machine, and back-end database or legacy machines.

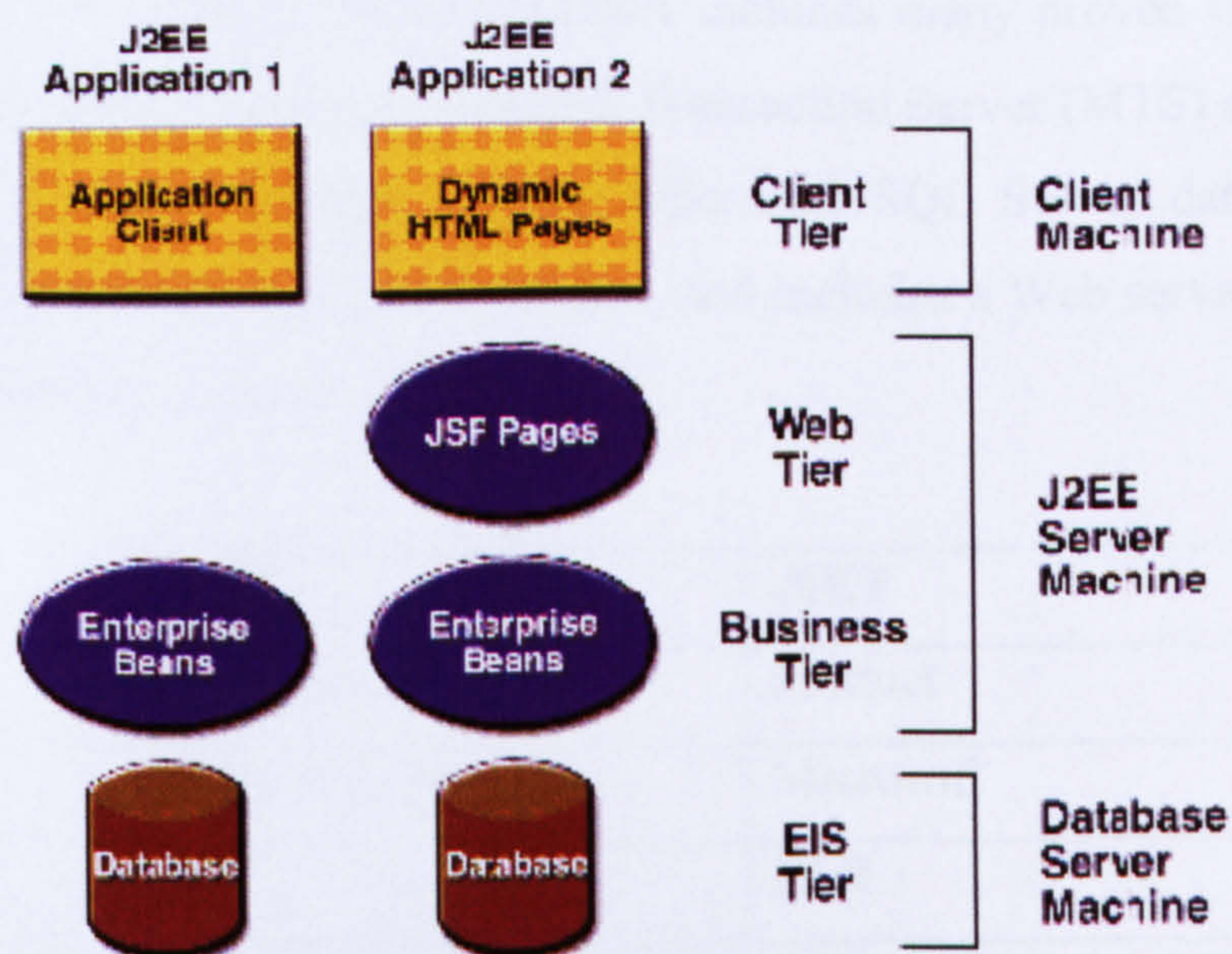


Figure 3.3 J2EE architecture (J2EE, 2004)

According to J2EE (2004), one of the main features of J2EE is its business component. Business logic that solves or meets the needs of a particular business domain (e.g. banking, finance) is handled by Enterprise JavaBeans (EJB) running in the business tier. J2EE specification ensures that the application developers are able to concentrate on the business logic programming, while the underlying services (e.g. security management, transaction management, resource pooling) are managed by the J2EE containers. Before a Web component, EJB, or application client component can be executed, it must be assembled into a J2EE module and deployed into its container. J2EE server provides



EJB container and Web container. Another significant change introduced by J2EE is how application components are packaged for deployment. In assembly process, J2EE components are packaged into modules (e.g. EJB JAR file, WAR file). Modules are then packaged into application (i.e. EAR file), which can be deployed on the application server. Each module contains a J2EE deployment descriptor which is an XML document providing instructions for deployment of the application (J2EE, 2004).

3.5.2 Microsoft .NET Platform

Microsoft .NET is largely a rewrite of Windows DNA (Distributed interNet applications Architecture), which was Microsoft’s previous platform for developing enterprise applications (Vawter et al, 2001). Windows DNA includes many proven technologies that are in production today, including Microsoft Transaction Server (MTS) and COM+, Microsoft Message Queue (MSMQ), and the Microsoft SQL Server database. The new .NET framework replaces these technologies, and includes a Web services layer as well as improved language support.

Feature	J2EE	.NET
Type of technology	Standard	Product
Middleware Vendors	More than 30	Microsoft
Interpreter	JRE	CLR
Dynamic Web Pages	JSP/Servlets	ASP.NET
Business Components	EJB	.NET Managed Components
Database Access	JDBC, SQL/J	ADO.NET
Web Services	Yes	Yes

Table 3.1 Comparison between J2EE and .Net

Table 3.1 is an analogy between J2EE and .NET which showcases the similarities. Both J2EE and .NET platform are powerful and can lead to the same destination. Choosing which platform to adopt has to consider existing developer skill sets, existing systems, existing vendor relationships, and customer requirements, etc.



### 3.5.3 PHP

PHP: Hypertext Preprocessor (PHP) (<http://www.php.net>) is an open source programming language, and used on the server to create dynamic Web pages. PHP was created in 1994 by Rasmus Lerdorf to track users at his web sites (Bakken, 2001). In 1995, Lerdorf released it as a package called “Personal Home Page Tools” (i.e. PHP2). PHP2 featured the built-in database support and HTML form handling. In 1997, two Israel developers, Zeev Suraski and Andi Gutmans rewrote the core engine of PHP and released it as PHP3, which substantially increased performance and led to an explosion in PHP use. The release of PHP 4, which featured the new *Zend Engine* and was much faster and much powerful than its predecessors, has further increased PHP’s usage. The latest version is PHP 5.0, which has built-in SOAP extension for interoperability with Web Services and adopts *Zend Engine II*, should further improve PHP’s popularity.

As a language that has been designed expressly for the Web, it brings many features that commercial entities are looking for (<http://www.php.net>):

- Exceptionally short learning curve
- Quick development time
- Very high performance

This is essential for companies who are faced with scarce of skilled programming resources and ever-tighter time to market deadlines. In addition, PHP supports all major platforms (UNIX, Windows and even mainframes), and features native support for most popular databases. All these factors make it a very good choice for Web development. Programmers familiar with languages such as C, C++ or Java frequently find that they can begin programming in PHP within a few hours. *Intranet Design Magazine* stated: "PHP was built with the needs of Web developers in mind... Unlike other cumbersome, overhead-laden approaches, PHP is lightweight and focused on the Web - where it can solve complex problem scenarios quicker and more easily than comparable technologies." (Intranet Design Magazine, 2004)



Java Server Pages (JSP) / Servlets is Sun Java's solution for delivering dynamic web pages. Like PHP, JSP is highly portable across different platforms (even though PHP supports any 32-bit or better platform, whereas JSP supports only platforms that have a Java virtual machine available). However, a beginner with no programming background will find it exceptionally difficult to begin working with JSP/Servlets due to the complexity of the language as well as the complicated JSP system design. PHP can integrate with Java, which means that it can be used as a scripting language for activating Java logic, just as much as JSP can. In addition, PHP can also integrate with other component models (e.g. Microsoft COM), and adding support for new object models is extremely easy. This means the extensibility of PHP is infinite (<http://www.php.net>). JSP, on the other hand, is completely dependent on Java.

### **3.6 State-of-Art Automatic Identification (Auto-ID) Technologies**

A number of state-of-art Automatic Identification (Auto-ID) technologies (e.g. RFID) can be employed to aid in dealing with information capture and dissemination accurately and efficiently during a product life cycle (Udoka, 1992). Auto-ID involves the automated retrieval of identity of objects and the information about the item to be stored, retrieved and communicated. Auto-ID systems have been used in industry for almost 20 years (McFarlane et al, 2003).

#### **3.6.1 Radio Frequency Identification (RFID) Technology**

Introducing RFID, the bar code technology has to be mentioned. One of the simplest forms of Auto-ID system is the device that contains some fixed information about the object that is associated with. The device may take the form of the tag that is physically attached to the associated object. Two common tagging technologies are barcode and RFID tag (Haller et al, 2002).



- **Barcode Technology**

Bar code technology is perhaps the most widely used Auto-ID technology that was developed in the early 1950's (Brown, 1997). The familiar black-and-white strips found today on most packaged goods contain the information describing the type of an item and the identity of its manufacturer. This information is communicated through the use of a bar code scanner, an optical device that can read such barcode when it is held in the reader's field of view.

The barcode standards are most commonly used for consumer item labelling in shops and supermarkets that identify the price, manufacturer and the Stock-Keeping Unit (SKU). Although barcode can theoretically be of any length, in practice the use of barcode is restricted in size and contains no serial number. Bar code technology requires shining light at it and looking at the intensity of the reflection. Both the transmitter and receiver components in light communication systems must have a direct, unobstructed straight-line path between them during communication (Brown, 1997; Haller et al, 2002).

- **Radio Frequency Identification (RFID) technology**

The RFID systems consist of Radio Frequency (RF) tags (or transponders) and RF tag readers (or transceivers). The transponder is located on the object to be identified and is the data carrier in the RFID systems. The transceiver may be able to both read and write data to the transponder. Transponders themselves typically consist of integrated circuits connected to an antenna (Finkenzeller, 2003). The use of silicon-based microchips enables a wide range of functionality to be integrated into the transponder. Transponders may be either active or passive. Active transponders have an on-tag power supply (e.g. battery) and actively send an RFG signal for communication, while the passive transponders obtain all of their power from the interrogation signal of the transceiver. The transceiver consists of a radio frequency module, a control unit, and a coupling element to interrogate electronic tags via radio frequency communication (Sarma, 2003; Finkenzeller, 2003; Haller et al, 2002). A basic RFID system is shown in Figure 3.4.



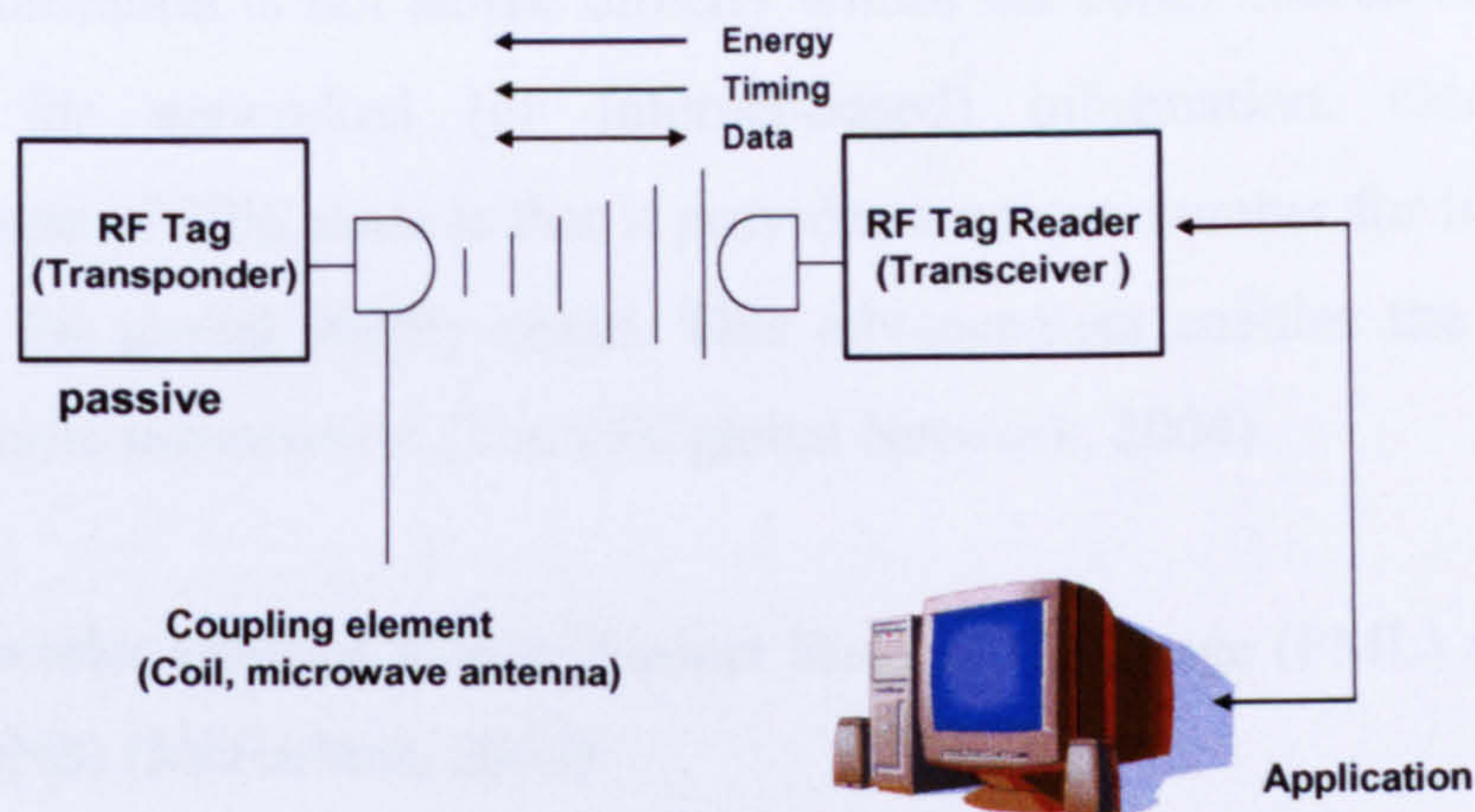


Figure 3.4 A basic RFID system (Finkenzeller, 2003)

The RFID tags have more storage capacity than barcode, and it is therefore possible to further partition the memory to include a serial number to uniquely identify the items. Furthermore, the radio communication works without line-of-sight, in some cases there can be quite substantial obstacles between the transmitter and receiver and communication is still viable. This is clearly a great advantage when the physical environment is less well constrained, and line-of-sight can not be guaranteed (Sarma, 2003; Finkenzeller, 2003; Haller et al, 2002).

One of the biggest disadvantages of RFID tag working in a passive way is the problem of sufficiently proper operation of stickers or labels mounted directly on the surface of metals. This will be used not only in machinery, but also in food labelling. There are some approaches to overcome this problem by improved shielding and tuning of the labels (Haller et al, 2002).

### 3.6.2 Electronic Product Code (EPC)

The Electronic Product Code (EPC) is a 96-bit code of numbers which is embedded in a memory chip, known as smart tag, on individual product (McFarlane, 2002). Each smart tag is scanned by RFID reader to identify the product embedded identity code. The 96-bit EPC code links to an online database, providing a secure way of sharing product-specific information along the supply chain (EPC used to identify item using RFID,



2004). Information is not stored directly within the code; instead the code serves as a reference for networked (or Internet-based) information. One of the critical advancements of EPC code is that it provides a unique number for individual objects in motion in the global supply chain. This advancement enables the communication of object-specific information (The EPCglobal Network, 2004).

The EPC works together with a Product Markup Language (PML) and Object Naming Service (ONS) (McFarlane, 2002).

### **3.6.3 Product Markup Language (PML)<sup>2</sup>**

The PML is a standard "language" for describing physical objects to the Internet (McFarlane, 2002). It is based on the eXtensible Markup Language (XML). As described previously, HTML is the common language on which most websites are based, allowing individuals to surf the Internet from their desktops. Where HTML tells a computer how information should be displayed (e.g. what color and size it should be), XML goes step further, telling the computer what kind of information it is viewing (e.g. an address or a telephone number) The PML will go even further, building in layers of increasingly specific data in order to describe physical objects, their configuration and state. It can be envisioned that potential use of PML includes (Kiritsis, 2004):

- Translate or contain static data, such as: dosage, shipping, expiration, advertising and recycling information.
- Provide instructions for machines that "process" or alter a product, such as: microwaves, laundry appliances, machine tools and industrial equipment.
- Present dynamic data: information that changes as a product ages or as it is consumed, such as: volume, temperature, moisture and pressure.
- Include software, or programs to describe how an object behaves. For instance: a PML file may contain the program which describes how fast the tires of car will wear before they need to be replaced, etc.

---

<sup>2</sup> At the time when the author amends the thesis after the viva, PML is no longer about EPC initiative, and EPC interface becomes a standard.



### 3.6.4 Object Naming Services (ONS)

The ONS tells computer systems where to locate the Internet about any object that carries the EPC code (McFarlane, 2002). ONS was developed at Massachusetts Institute of Technology by Dr. David Brock, Professor Sanjay Sarma and Joseph Foley. ONS is based on in part on the Internet Domain Name System (DNS), which routes the user request information to appropriate web site over the Internet. It is likely that the ONS will be many times larger than DNS. The ONS system is capable of quickly locating data for every single one of the trillions of objects identified by EPC (McFarlane, 2002).

## 3.7 Summary

There is a vast array of enabling technologies have to be examined in order to identify most appropriate ones for this research. As a result of this survey, the following ICT were chosen to be employed within this research study:

- 1) *Web-enabled Information Management System (IMS)* to prototype service enablers to support the ELIMA field trials - Currently most information management systems are Web-based, and integrated with DBMS and data management functionalities and tools. Hence they can be used to effectively manage the product life cycle data in a systematic and integrated way.
- 2) *Entity-Relationship (E-R) modelling* for life cycle database analysis and design- As the lifecycle database should be designed to house product life cycle data, various stakeholders' data and system data, etc; should be designed to be robust and scalable; should support data interoperability, etc, the adoption of E-R modelling for the life cycle database analysis and design can ensure these requirements are met.
- 3) *Relational Database Management System (RDBMS)* for life cycle data management - In comparison with OODBMS, RDBMS is more popular and



there are a range of relational DBMS products available, e.g. MySQL, MicroSoft SQL Server, IBM DB2 and Oracle, etc.

- 4) *Unified Software Development Process* (USDP) approach for the service enabler framework development - USDP emphasises iterative and incremental software development and incorporates UML. It also comprises much good common practice of software development. USDP is now being widely used.
- 5) *Object-Oriented* (OO) approach for the service enabler framework analysis and design - OO approach now dominates the analysis, design and implementation of information systems and software applications.
- 6) *Unified Modelling Language* (UML) for system and software modelling - OO approach adopts UML as a modelling language.
- 7) *eXtensible Markup Language* (XML) for life cycle data presentation, communication and exchange - XML is a *de facto* standard for data presentation, communication and exchange.
- 8) *Java* and *PHP* as a Web programming languages - Both *Java* and *PHP* are free and widely used in academic research. PHP is lightweight and powerful.



## Chapter 4 A Scheme of Enabling Services from Product Life Cycle Data

### 4.1 Introduction

In this chapter, a scheme which actively utilises product life cycle data in a systematic and integrated manner to facilitate the creation and delivery of suitable services during the life cycle of a product is proposed inspired initially by ELIMA project. The scheme consists of (i) A product embedded with IDU (ii) Communication Support Infrastructure (CSI) and (iii) a service enabler which is a collection of suitable software components capable of receiving product life cycle data and using it to provide integrated information and knowledge which can be used by service providers. The conceptual diagram for the scheme is shown Figure 4.1.

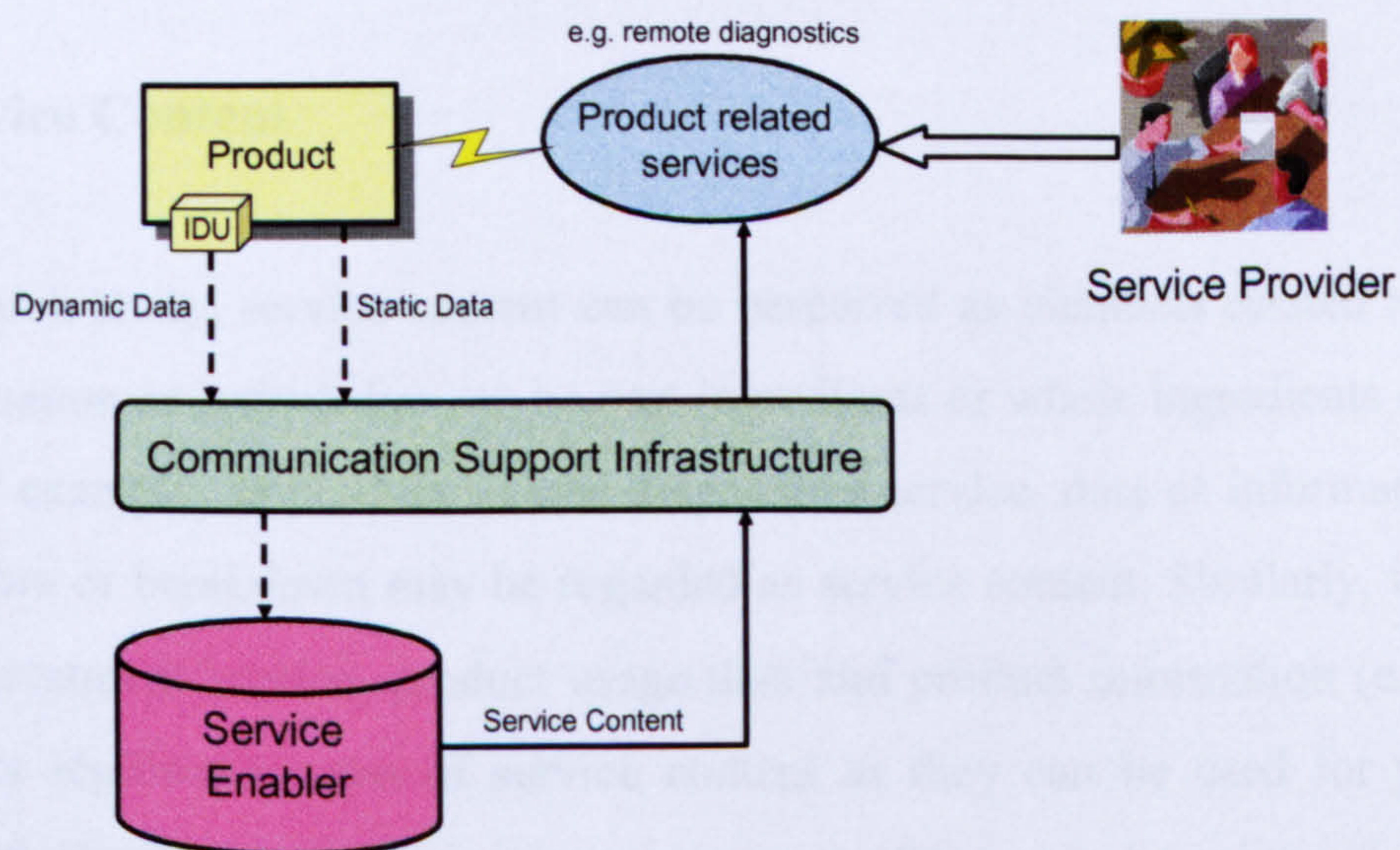


Figure 4.1 Service enabling scheme conceptual diagram

In this context, the chapter consists of following main sections:

- (i) Service and Service Content
- (ii) Rationale of Enabling Services from Product Life Cycle Data
- (iii) Service Enabling



## **4.2 Service and Service Content**

### **4.2.1 Service Definition**

A service can be perceived as a non-material and perishable activity or process (Mont, 2000; Tomiyama, 2001). Service normally means an activity of economic value done for third parties, usually on a commercial basis (e.g. rental, maintenance) (Wong, 2003). Ramaswamy (Ramaswamy, 1996) defines service as a business transaction that takes place between a service provider and a customer in order to produce an outcome that satisfies the customer. Broadly speaking, within a manufacturing enterprise, a service (e.g. use pattern analysis) provided by one department (e.g. marketing department) to another department (e.g. design & development department) can also be treated as a service although the commercial value may not be explicit. In this research study, the latter meaning of service is adopted.

### **4.2.2 Service Content**

In this research study, service content can be perceived as elements related to service. Data, information or knowledge can be part ingredients or whole ingredients of service content. For example, in product remote diagnostics service, data or information about product failure or breakdown may be regarded as service content. Similarly, in product end-of-life treatment service, product usage data and product information (e.g. Bill of Material) are legitimate items of service content as they can be used for predicting remaining life time of product/components/parts, etc. Thus any data element which has a potential of assisting the actual service may be regarded as part of the service content.

## **4.3 Rationale of Enabling Services from Product Life Cycle Data**

The rationale of enabling services from product life cycle data is that: (i) life cycle data is valuable asset (ii) information and knowledge can be obtained and discovered from data by using statistical analysis and intelligent data mining techniques (iii) Data, information and knowledge can enable appropriate services to be delivered effectively and efficiently.



### 4.3.1 The Value of Product Life Cycle Data

Once data has been reliably captured over a suitable period of time, it can yield information in major areas of interest such as (Simon et al, 1998; Simon et al, 2000; Moore et al, 2000):

- *Marketing*: Use pattern analysis through life cycle data can be used to improve the design of the product and to gain marketing advantage from the improved knowledge of user's behaviour.
- *Reliability*: Data on failure modes and frequencies can show how design can be varied to improve reliability.
- *Servicing*: Data can be used for remote diagnostics and monitoring.
- *Preventive Maintenance*: Data can be used for condition monitoring, avoiding breakdown and allowing convenient, cost-saving maintenance instead.
- *Warranty*: Warranty can be based on actual use rather than a time period. This can help reduce cost.
- *End-of-life treatment*: Historical data can be used to predict remaining life span of the parts/subassemblies/material.

In addition, Kiritsis (2004) in his "*Ubiquitous Product Lifecycle Management using Product Embedded Information Devices*" (*PROMISE* project) states that data gathered during a product life cycle can be used for decision making. The *PROMISE* project will develop appropriate technologies, including product lifecycle models, product embedded information devices with associated firmware and software components and tools to acquire and process product life cycle data.

### 4.3.2 Knowledge Discovery

In order to translate the raw data into meaningful trends, hidden patterns and to discover knowledge, appropriate statistical analysis and data mining is required.



Data mining refers to the mining or discovery of new information in terms of patterns or rules from vast amount of data (Connolly et al, 2002; Elmasri et al, 2000; Atzeni et al, 1999), as shown in Figure 4.2. Thus data representing factual information (e.g. measurements or statistical data based on measurements) can be used as a basis for knowledge discovery.

According to Elmasri et al (2000), the goal of data mining includes:

- *Prediction*: Data mining can show how certain attributes within the data will behave in the future.
- *Identification*: Data mining can be used to identify the existence of an item, an event, or an activity.
- *Classification*: Data mining can partition the data so that different classes or categories can be identified based on combination of parameters.
- *Optimisation*: The eventual goal of data mining may be to optimise the use of limited resources such as time, space, money, or natural resources and to maximum output variables such as sales or profits under a given set of constraints.

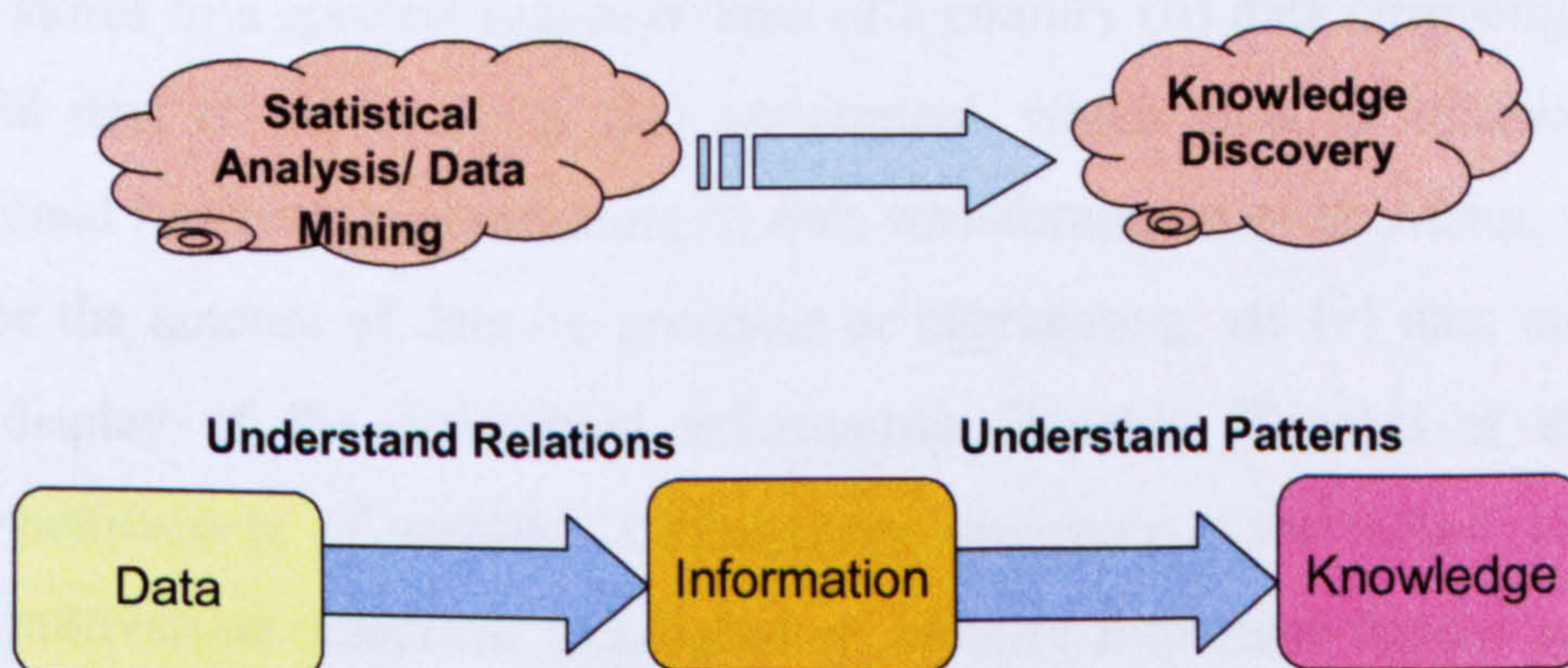


Figure 4.2 Data, information and knowledge transformation (Connolly et al, 2002; Elmasri et al, 2000; Atzeni et al, 1999)

In recent years, the application of statistical analysis and data mining techniques to find and interpret patterns from data has generated extraordinary interest in the business community (Zanakis et al, 2004). The statistical analysis and intelligent data mining techniques have been widely utilised in the scientific and engineering research fields,



for example in breast cancer diagnosis (Kovalerchuck et al.,2000; Delen et al,2004), sheet metal assembly variation diagnosis (Lian et al,2002), the discovery of unusual folding regions in genome sequence (Le et al, 2001 ) etc. Availability of cheap and powerful computational power through standard computers has accelerated this trend.

The pervasive ICT have contributed to the increasing popularity of statistical analysis and data mining applications in business concerns. For example, database and information management system can be used to house and manage the product life cycle data in an integrated and systematic manner; Internet and various Web technologies (e.g. Java, XML) are used for data communication and presentation; various data mining software packages (e.g. Excel, SPSS) can be employed for discovering the pattern and relationships.

A technique called Knowledge Discovery in Database (KDD) (Apte et al, 1996) used to find and interpret patterns from database is now well recognised. The knowledge discovery process of KDD comprises six phases (Adriaans et al,1996): (i) data selection, which concerns the selection of data about specific items or categories, or from stores in a specific region or area of a country (ii) data cleansing, which corrects invalid data or eliminates it (iii) enrichment, which aims to enhance the data with additional sources of information (iv) data transformation or encoding, which is used to reduce the amount of data by grouping or aggregating, etc (v) data mining (vi) report and display of the discovered information. Zanakis (Zanakis et al, 2004) in the “*Competitiveness of nations: A knowledge discovery examination*” used data mining and multivariate statistical techniques to identify important factors associated with a country’s competitiveness, and developed KDD models to predict it.

### 4.3.3 Data, Information and Knowledge Can Enable Services

The importance of data, information and knowledge in the delivery of efficient services is being investigated in several projects. For example, in a UK government DTI (Department of Trade and Industries) funded project, namely, *Equipment Management Trial*, under the Next Wave Technologies and Markets Programme (“Next Wave”)



(<http://www.nextwave.org.uk>), various equipments are connected to the home network hence there is considerable data flow within the network. Capturing, recording and processing this data by employing ICT can open up new dimensions for equipment management (Chong et al, 2004). Within the home, services such as enhanced operational monitoring and remote diagnostic are anticipated. Various tasks could be scheduled to be carried out automatically, and users could be kept informed of the equipment status and alerted at the time of decision making or emergency.

The importance is also being examined in “*eServices-on-Demand for connected homes*” project, which is currently being conducted by De Montfort University (<http://www.mrg.dmu.ac.uk>) and OpenHub Ltd (<http://www.openhub.co.uk>). This project is to investigate innovative ways to deliver services (i.e. healthcare and security services) to connected homes by employing ICT. In this project, for example, the customer’s profile data (e.g. age, sex, height) and health data (e.g. weight) is collected to calculate the Body Mass Index (BMI), which can indicate whether the customer is underweight, normal or overweight. With the information of BMI, the food recipe service is enabled thus the health nutrition consultant can provide recipe service to the customer (e-Service on Demand, 2005).

## 4.4 Service Enabling

### 4.4.1 Service Enabling Scheme

A more detailed view of the proposed scheme is illustrated in Figure 4.3.

The operation of the scheme relies on the dynamic data acquisition which is achieved through IDU. Each individual product is equally equipped with an IDU which has sufficient storage capacity and has the ability to transmit the acquired life cycle data to the specified destination. A communication support infrastructure is required which provides a communication channel for various flows such as static data flow, dynamic data flow, service content flow, etc according to the agreed protocols. A service enabler lies at the heart of the system. It is a collection of suitable software components such as database, knowledge base, information engine and inference engine, etc capable of



receiving product life cycle data and using it to provide integrated information and knowledge which can be used by service providers. Details of the service enabler will be developed in later sections.

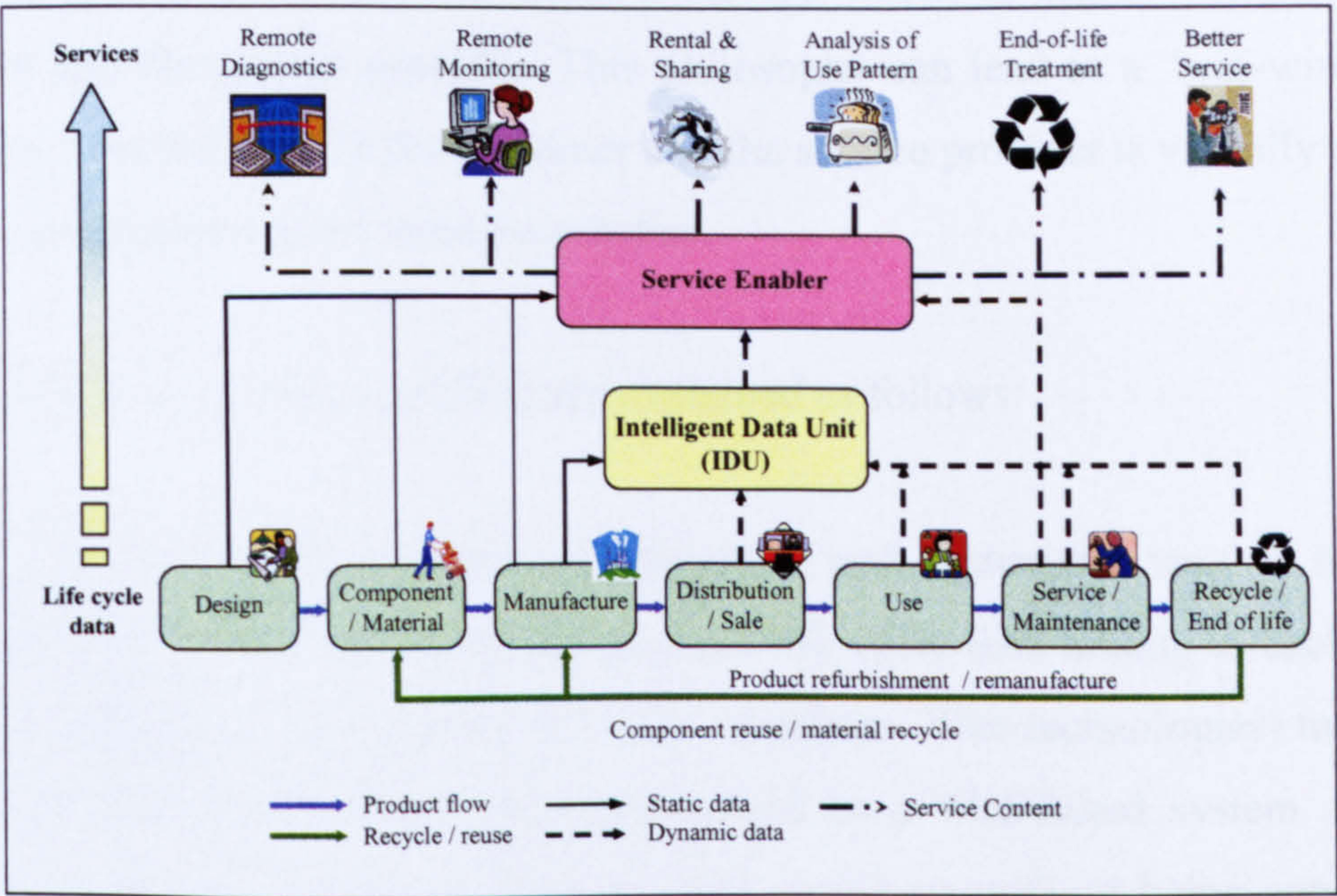


Figure 4.3 Schematic presentation of service enabling scheme

The service enabler plays an extremely important role in the scheme. Functionally, the service enabler:

- Receives the life cycle data which starts from the product design stage through to the recycle/end-of-life stage.
- Stores and manages the life cycle data in a systematic and integrated way.
- Creates appropriate service content by utilising suitable business logic and algorithms

#### 4.4.2 Service Enabler

The service enabler is defined in this research study as “an intelligent information management system or expert system or combination of both that is capable of creating service content from product life cycle data to facilitate the creation and delivery of appropriate product related services during a product life cycle“. The service enabler can be perceived as a form of service engine. It is a collection of suitable software



components such as database, knowledge base, information engine, and inference engine, etc. The input to the service enabler is the raw product life cycle data and the output is an array of intelligence and service content which can be used by service providers to deliver to the end users a service which is highly beneficial to both customer and the service provider. This philosophy can lead to a “win-win” situation where any tension between the customer and the service provider is virtually eliminated, and hence strengthen good business practice.

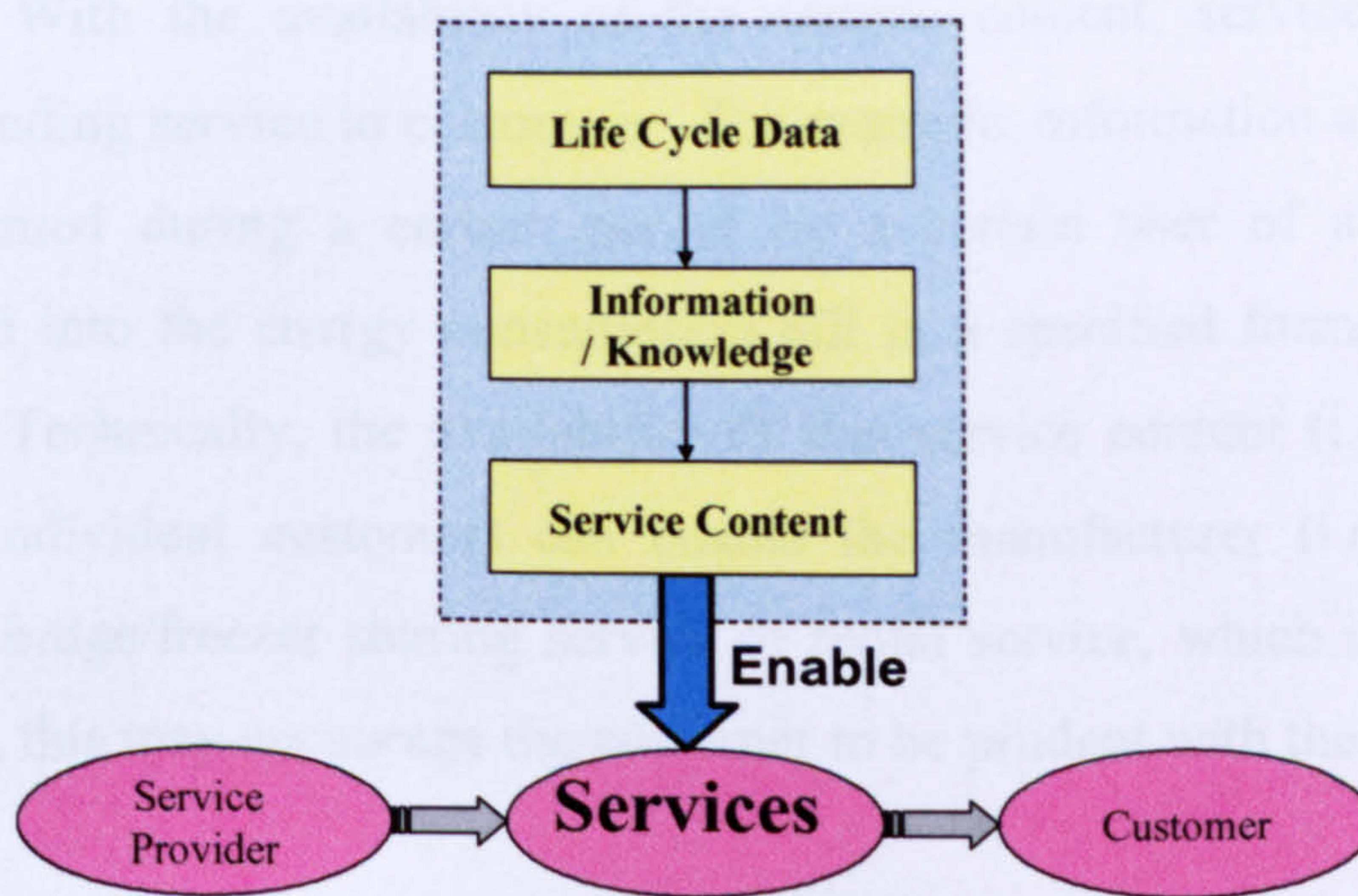
Some aspects of the service enabler are explained as follows:

- 1) The service enabler provides an integrated and systematic way to manage the product life cycle data. It houses product life cycle data arising at each life cycle stage and employs appropriate ICT (e.g. database, Web technologies) to manage it. The service enabler should be implemented as a Web-based system so that any authorised users can access the data from anywhere without being constrained by any geographical barrier.
- 2) The service enabler is distinct from the service provider. Service enabler is mainly concerned with enabling services from the gathered life cycle data, while the service provider is responsible for the actual delivery of the enabled services to customers, or end-users.
- 3) Service enabler can be an individual information management system, or individual expert system or both. If the expert system is employed in the service enabler, the knowledge set (e.g. expertise, knowledge) which is generally available from books, magazines and knowledgeable persons should be put into the knowledge base within the service enabler beforehand (Sell, 1985; Durkin, 1994).

#### **4.4.3 Service Enabling Process**

The overall process of enabling services from product life cycle data is illustrated in Figure 4.4.





*Figure 4.4 Service enabling process*

The service enabling process is through the following 3 steps. Each step involves employing appropriate ICT to complete the task.

Step 1 - Receive life cycle data. Life cycle data can be in various data format (e.g. ASCII file, data stream). It can be transferred to service enabler via various communication protocols (e.g. TCP/IP, GSM) in different manner. For example, the gathered dynamic data can be saved as a local file (e.g. ASCII file, XML document) which is then uploaded to service enabler through TCP/IP.

Step 2 - Obtain the information and discover the knowledge from the raw life cycle data by using statistical analysis and data mining techniques. For example, using some simple statistical analysis such as sum, averaging, frequency distribution, etc, meaningful information or trends can be obtained to help with the decision making. The utilisation of ICT can facilitate the management of large amounts of data, computation of complex applications and large-scale quantitative analysis with high performance and improved accuracy.

Step 3 - Package or integrate information/knowledge into service content. The generated information and knowledge need to be configured or transformed into a template or format usually pre-defined by the service provider to form the appropriate service



content. With the availability of the service content, service provider can deliver corresponding service to customers. For example, information about how much energy is consumed during a certain period by a certain user of a fridge/freezer can be packaged into the energy consumption bill in a specified format to formulate service content. Technically, the availability of this service content (i.e. energy consumption bill of individual customer) can enable the manufacturer (i.e. service provider) to provide fridge/freezer sharing service or rental service, which is promoted by PSS. In addition, this may encourage the customer to be prudent with the energy consumption.

## 4.5 Summary

The service enabling scheme was proposed as an integrated mechanism to enable appropriate services (e.g. remote diagnostics) to be delivered effectively and efficiently during a product life cycle. It consists of a product embedded with IDU, communication support infrastructure and service enabler.

The service enabler lies at the heart of scheme. It is a collection of suitable software components such as database, knowledge base, information engine, and inference engine, etc capable of receiving product life cycle data and using it to provide integrated information and knowledge, or appropriate service content for the benefit of service providers.



## Chapter 5 The Realisation of Product Service Systems for Consumer Products

### 5.1 Introduction

The proposed service enabling scheme lays a foundation for realising PSS for consumer products. In this chapter, a methodology for realising PSS for consumer products is proposed. The whole process of developing this methodology is illustrated as follows: (Figure 5.1).

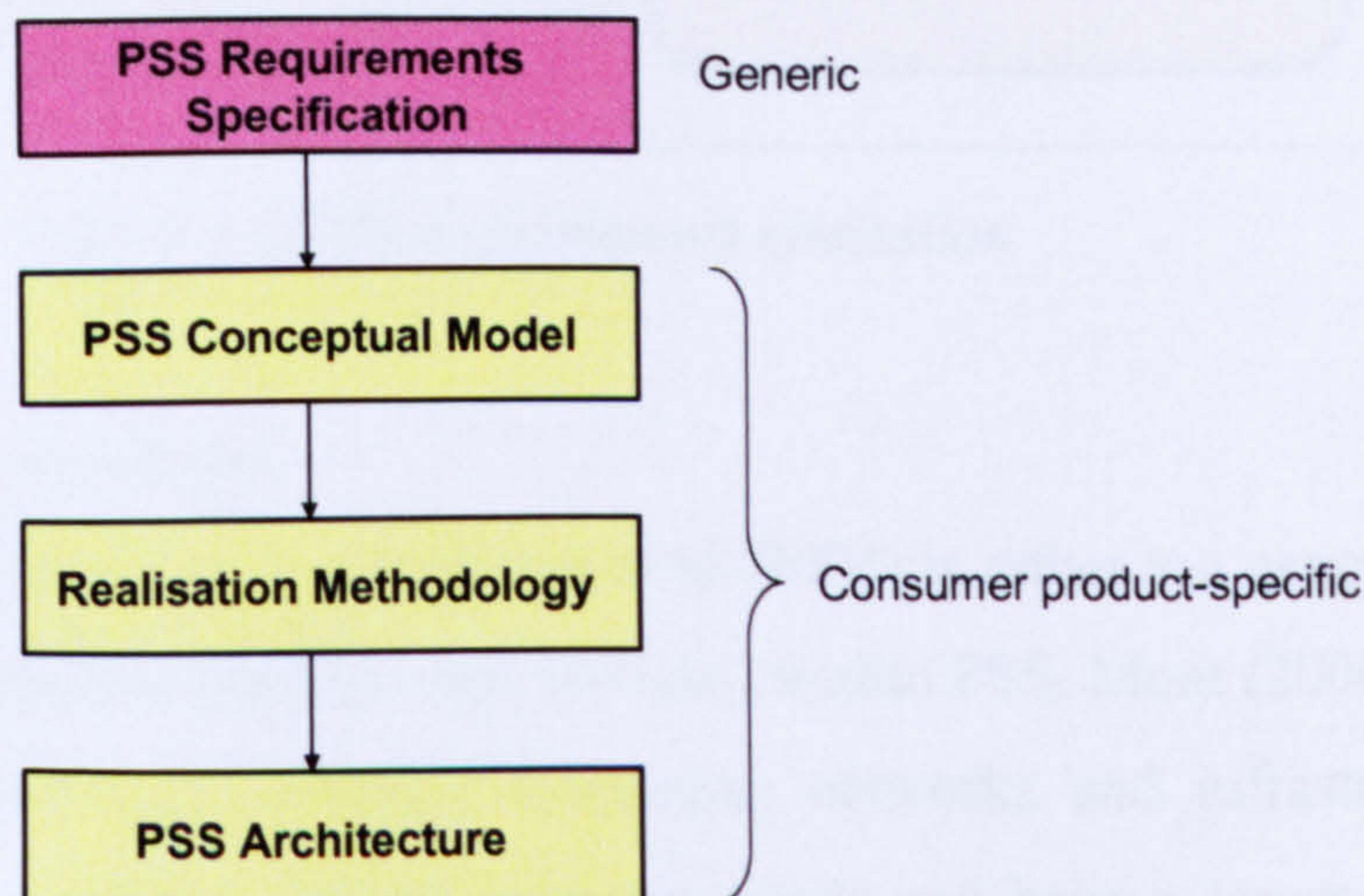


Figure 5.1 Development process of PSS realisation methodology

- (i) Develop a generic PSS requirements specification.
- (ii) Propose a PSS conceptual model and methodology based on the PSS requirements specification and service enabling scheme.
- (iii) Develop PSS architecture

This chapter has the following main sections:

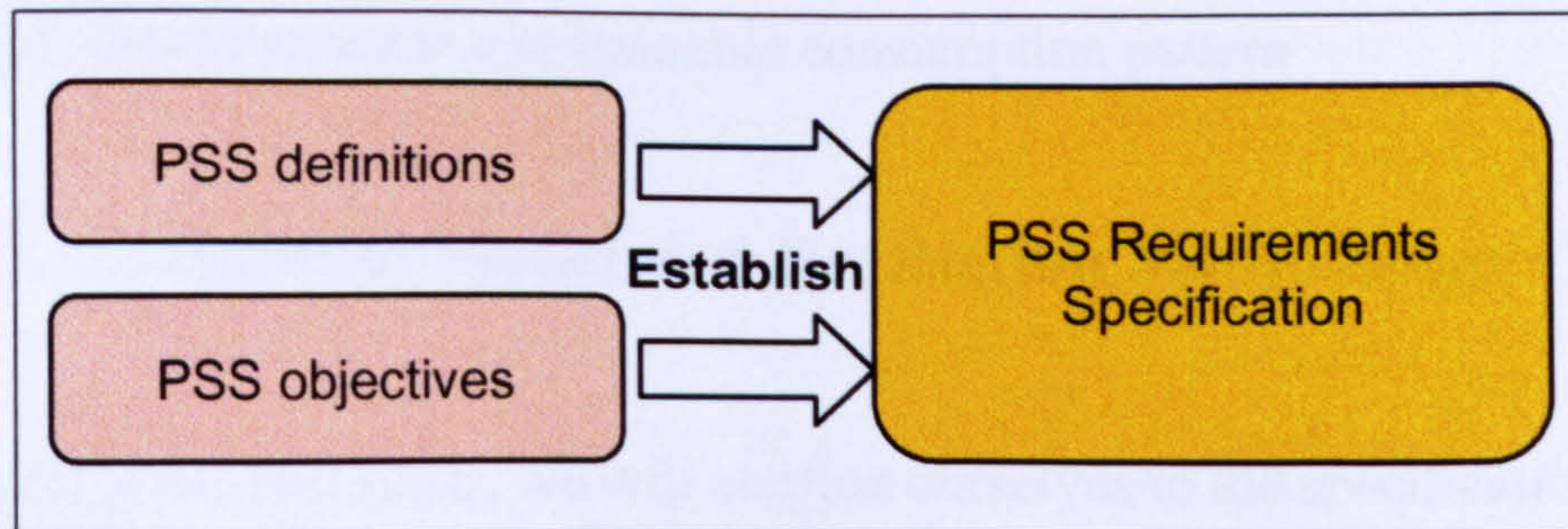
- (i) PSS Requirements Analysis
- (ii) Conceptual Model and Realisation Methodology
- (iii) A Typical Implementation of PSS for Consumer Products
- (iv) PSS Architecture
- (v) The Realisation Methodology and ELIMA Project



## 5.2 PSS Requirements Analysis

### 5.2.1 PSS Requirements Elicitation

The process outline of capturing PSS requirements is introduced as follows (Figure 5.2).



*Figure 5.2 PSS requirements elicitation*

- Review of PSS definitions

The common PSS definition by Goedkoop et al (2003) is rather too general and defines only two components (i.e. products and services) within PSS. Mont (2000) defined PSS as “a system of products, services, supporting networks and infrastructure that is designed to be: competitive, satisfy customer needs and have a lower environmental impact than traditional business models”. This is a more helpful definition and four major constituents of PSS have been identified: products, services, supporting networks and necessary infrastructure. But how the services can be delivered is not entirely clear.

- Review of PSS objectives

As a reminder, the objectives of PSS can be summarised as follows:

- to deliver a product service mix
- to promote a new business model of a sustainable consumption pattern - “have a utility instead of an ownership”

### 5.2.2 PSS Requirements Specification

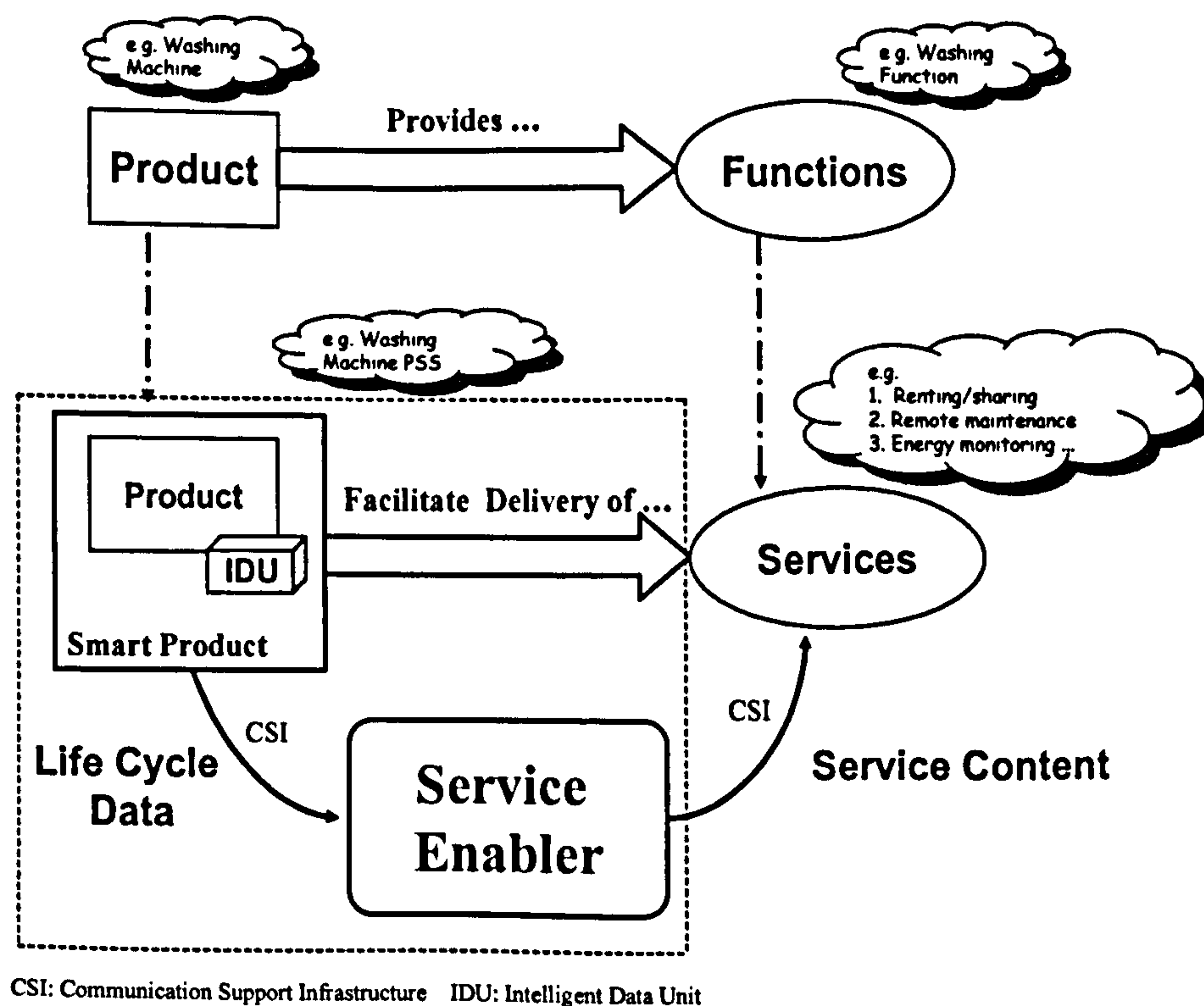
Considering the above definitions and objectives, a generic specification of PSS may be expressed as follows:



- PSS must contain two compulsory elements, i.e. product and services
- PSS should have a capability to facilitate the creation and delivery of appropriate product related services
- PSS should have a communication-supporting network so that services can be delivered
- PSS should promote a sustainable consumption pattern

### 5.3 PSS Conceptual Model and Realisation Methodology

For the purpose of this thesis, we will confine ourselves to the specification which relate to the product-oriented PSS and use-oriented PSS for consumer products. Based on the general PSS requirement specification and the proposed service enabling scheme, we developed a relevant conceptual model, as shown in Figure 5.3.



**Figure 5.3** *A conceptual model for product-oriented PSS and use-oriented PSS*

The PSS conceptual model for consumer products suggests a PSS realisation approach as follows:



### (1) Design of IDU

By integrating the IDU with the product, the product became a smart product which has the in-built functionality to automatically acquire dynamic data and make it available to all stakeholders involved during the product life cycle (Kango et al, 2003). Since IDU is likely to be the major source of product life cycle data and likely to be automated in data transmission, the design is very important.

The design of IDU is product-dependent and involves technologies such as mechanical engineering and electric engineering, etc. Since these technologies have been well established so far, and IDU has also been well developed, details of IDU design will not be included in this research.

### (2) Transmission of life cycle data

The transmission of the data can be automated according to the agreed protocols of communication, or in other forms of transmission. The criteria for choosing appropriate communication technologies and protocols include distance, speed, bandwidth and size of data, etc. Vast expertise in this area exists, for example, Internet and wireless communication (i.e. GSM) become ubiquitous nowadays and they are usually chosen for life cycle data transmission. Moreover, there are many other standard communication technologies (e.g. Bluetooth, Lonworks) and many “plug-and-play” communication components. Hence the thesis will not go into the details of the transmission of life cycle data.

### (3) Development of service enabler

This research concentrates on the design and development of service enabler, due to the following reasons:

- Service enabler is proposed in this research as up till now there is little experience and know-how which can be used for reference.



- Service enabler is a core building block in this PSS solution

The PSS for consumer products is realised in such a way that the information loop in different stages of life cycle can be closed and a product service mix can be delivered. Moreover, the PSS conceptual model indicates a distinction between traditional consumer product and PSS. Traditionally, a product is associated only with its basic functionality (e.g. washing machine provides a washing capability). However with PSS philosophy, the product becomes part of the chain which delivers many other services in addition to basic functionality.

## 5.4 A Typical Implementation of PSS for Consumer Products

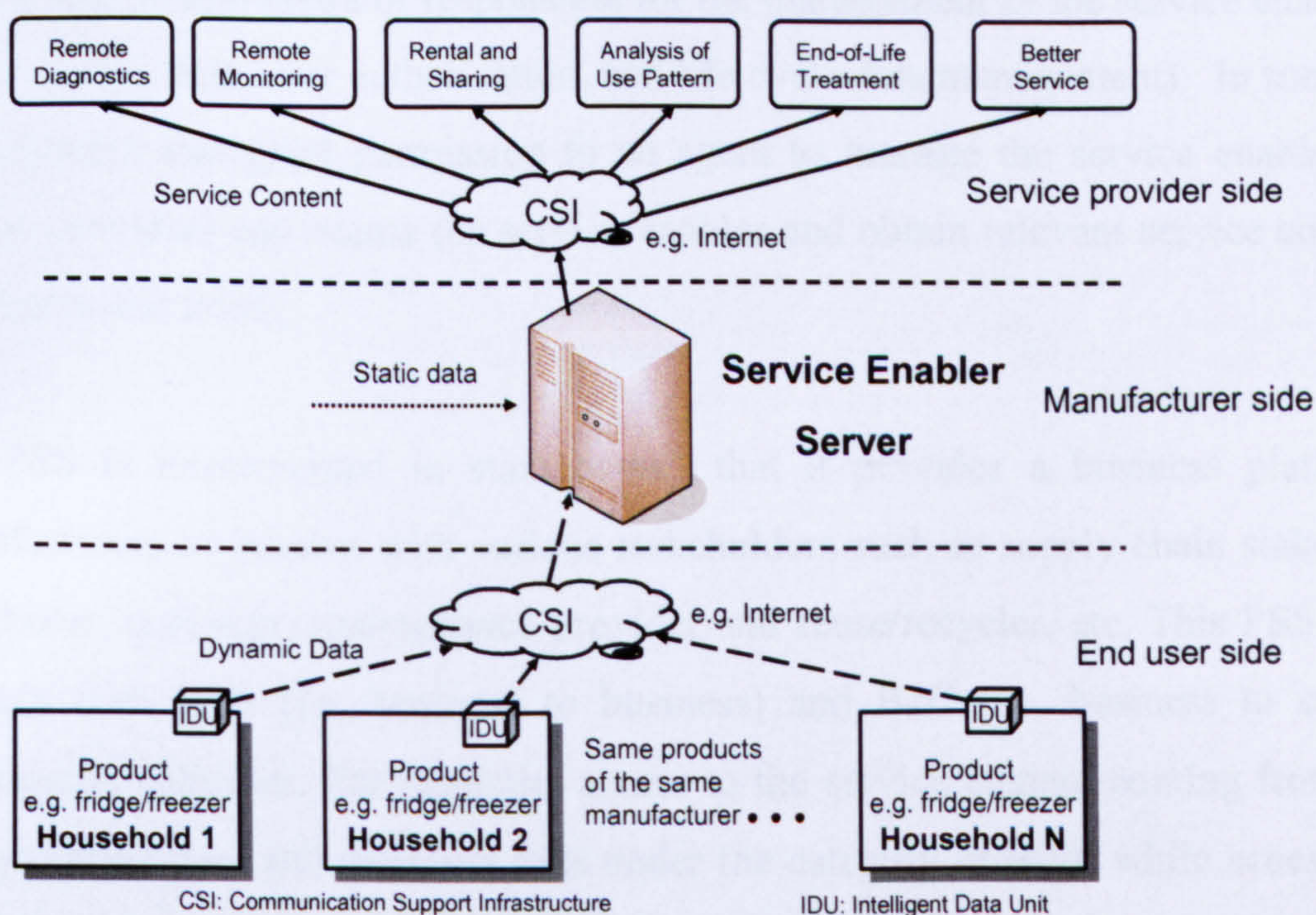


Figure 5.4 A typical implementation of PSS for consumer products

A typical implementation of PSS utilising the proposed realisation methodology is illustrated in Figure 5.4.

The PSS implementation for consumer products is a composition of hardware (e.g. IDU), software (e.g. service enabler system), communication facilities (e.g. Internet), and suitable management tools (e.g. Internet management) all integrated to enable appropriate services. In terms of implementation and deployment, the physical service



enabler can be a high-performance central server situated at the manufacturer site. Life cycle data acquired from a set of similar products via corresponding IDUs is transmitted to the central server over the wide area networks (e.g. Broadband) to create service content. The central service enabler houses both data and service content for different end users. Since each product has a unique ID, the life cycle data and the created service content are identified by this ID so that they can be distinct from others. In terms of security, only the authorised service providers or stakeholders such as supply chain stakeholders, recyclers, and customers, etc. can have privileges to access the central server to obtain the required service content.

The service enabler is usually situated at the manufacturer's site. The manufacturer plays a role of administrator responsible for the management of the service enabler (e.g. input of static data, user authorisation, and life cycle data management). In some cases, manufacturer can grant permission to an agent to manage the service enabler. Thus, service providers can access the service enabler and obtain relevant service content for their particular needs.

The PSS is implemented in such a way that it provides a business platform for manufacturers to interact with various stakeholders such as supply chain stakeholders, distributor, customer, maintenance provider and reuse/recycler, etc. This PSS solution features both B2B (i.e. business to business) and B2C (i.e. business to customer) commercial activities. For example, access to the service content coming from supply chain stakeholders and recyclers falls under the category of B2B; while access to data and information coming from end-users falls under B2C.

## **5.5 PSS Architecture**

The conceptual model for product-oriented PSS and use-oriented PSS has defined the system boundary and some internal components. However, the PSS architecture, especially the architecture of the core component - the service enabler - requires further detail. In order to accommodate this need, an architecture for both product-oriented PSS and use-oriented PSS has been developed with reference to the service enabling scheme



and the identified PSS requirements. It can be considered as a blueprint based on which a sustainable PSS solution for consumer products can be implemented and realised.

### 5.5.1 PSS Architecture Overview

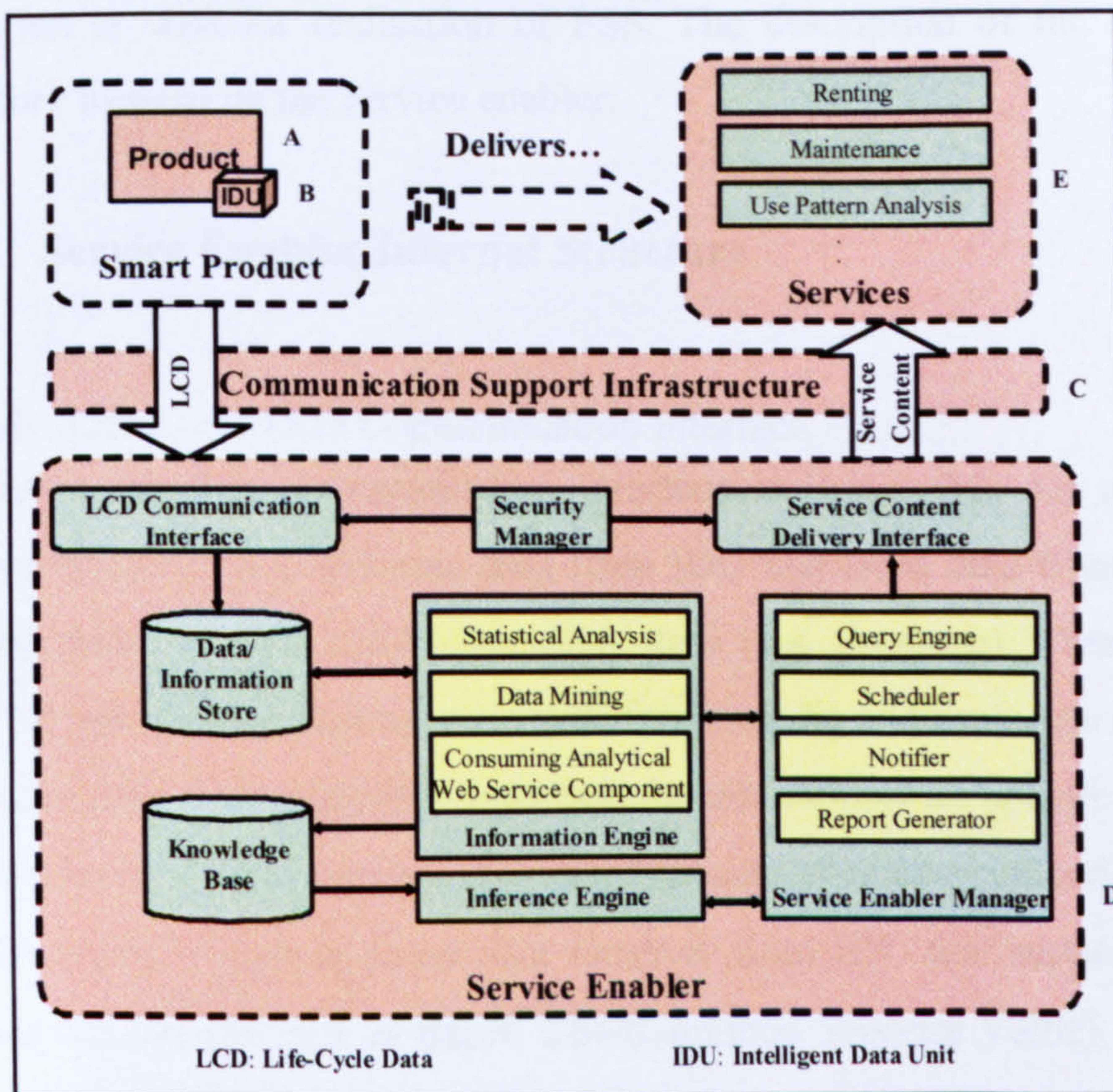


Figure 5.5 An architecture for product-oriented PSS and use-oriented PSS

The proposed PSS architecture is shown in Figure 5.5. The architecture presents internal constructions and component modules (e.g. service enabler), and indicates paths of interaction amongst those component modules and sub-modules. The PSS consist of five modules (as shown in the Figure 5.5 A, B, C, D, E), namely, the product, IDU, CSI, service enabler and services. The service enabler consists of eight sub modules:

- Communication interface
- Service content delivery interface
- Data/information store
- Knowledge base
- Information engine
- Inference engine



- Security manager
- Service enabler manager

The service enabler has been identified as a core component, hence its structure and operation is vital for realisation of PSS. The description of the PSS architecture is therefore focused on the service enabler.

## **5.5.2 Service Enabler Internal Structure**

### **5.5.2.1 Life Cycle Data Communication Interface**

The life cycle data communication interface is responsible for receiving data from various channels (e.g. dynamic data from IDU and BoM data from manufacturer) and feeding them into the data/information store (e.g. database). Therefore, the interface must be capable to communicate via different media and protocols to acquire data. The life cycle data communication interface should: support an upload utility to streamline the data transfer from the data collector (via auto or manual upload mechanisms) to the service enabler; support direct data retrieval from IDU and support data presented in various file formats such as Excel, CSV(Common Separate Value), XML, etc. As there are a broad spectrum of data sources and information types involved in the service enabler, the service enabler should incorporate the well-recognised open standard, i.e. XML, as the format for the data presentation and exchange.

### **5.5.2.2 Service Content Delivery Interface**

The service content delivery interface is a user-oriented interface through which service content is delivered. Access to the generated service content can be via a variety of channels (e.g. Internet, dial-up, LAN), operating devices (e.g. mobile phone, PDA and computer) and operating platforms (e.g. Palm, Microsoft Windows and UNIX). A unified or common look and feel interface for users would assist the utilisation of the service enabler. Furthermore, the interface should support various service providers or stakeholders to access the service content locally or remotely. A web browser-based user interface has been recommended because of its familiarity to most users.



### 5.5.2.3 Security Manager

Security is paramount to the service enabler due to the commercial value of the data it holds and the requirements of the Data Protection Act. A security manager is responsible for access control to the service enabler, to avoid any unauthorised access. In practice each user, including the IDU, has to establish an account and to log-on to the account before the access to the service enabler is granted. Each user account is associated with a security profile, which is a collection of security-related information (e.g. access privilege level) stored in a secure database. The security manager employs this information to authenticate the identity of the user.

### 5.5.2.4 Data/Information Store

Data/information store not only provides a persistent storage for life cycle data, but also provides all the essential housekeeping functions to enable other functional modules (e.g. information engine, service enabler manager) to manage and provide access to the stored data. In most cases, database management system (DBMS) is used as the data/information store. XML can also be used as the data/information store, but its supporting housekeeping functions such as data sorting, data query are not as robust and flexible as those of a DBMS. The diagram of Figure 5.6 shows the main functional blocks of the DBMS.

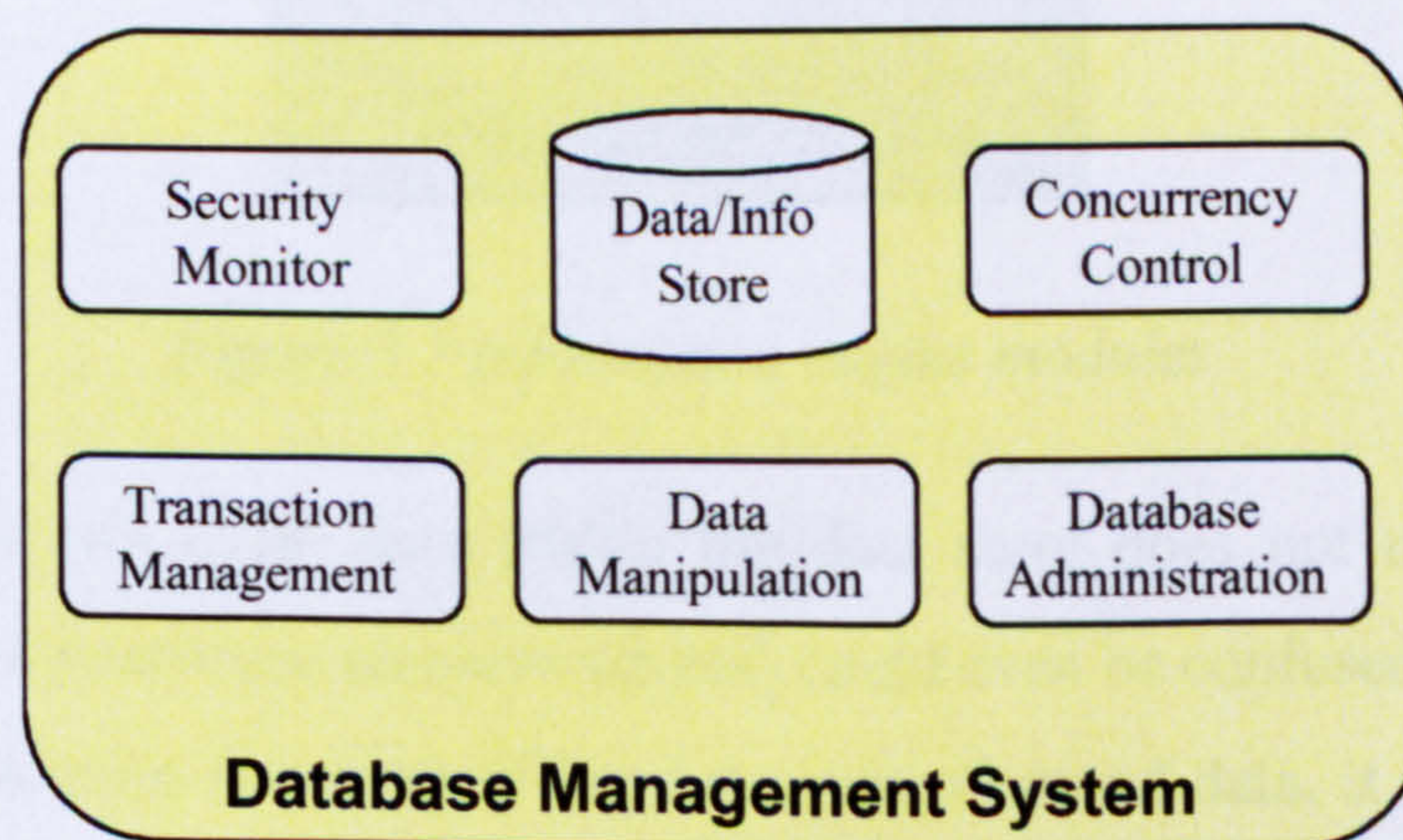


Figure 5.6 Database management system (Rolland, 1998)

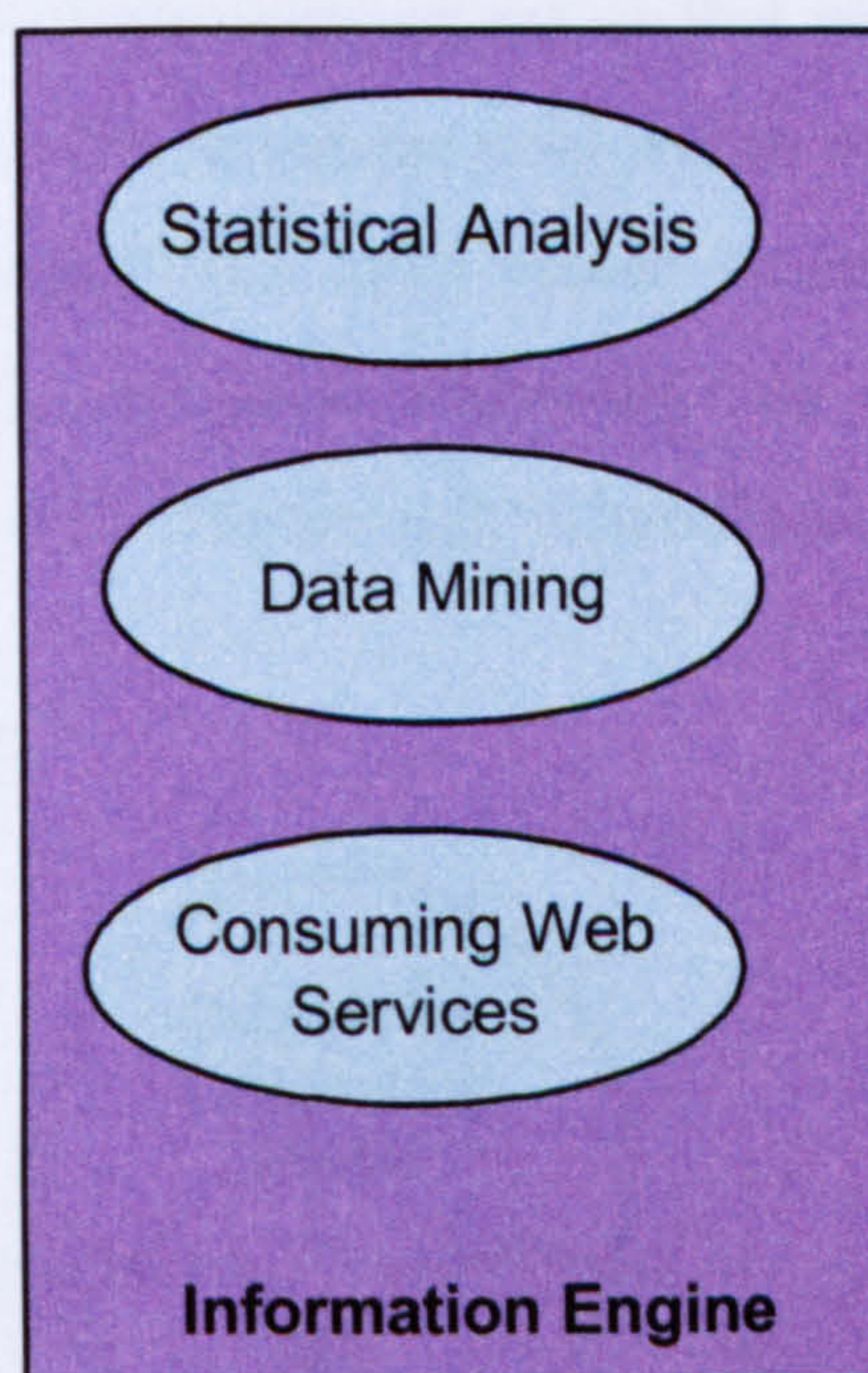
The data/information store for the service enabler should:

- Support client-server architecture by facilitating the implementation of Intranet and Internet access of the data/information residing in the central repository



- Provide functionality for storing, retrieving, modifying, deleting and browsing the data/information
- Provide persistent storage to house the data/information in various forms, for example, text, binary and graphics
- Provide the transaction management for the service enabler
- Support both intuitive and structured query methods to retrieve data/information
- Support information integration from multiple sources

#### 5.5.2.5 Information Engine



*Figure 5.7 Information engine modules*

Raw product life cycle data within the data store does not necessarily offer many benefits. In a worst-case scenario actor(s) could even be confused due to the mass of the data. To maximise the value of large amount of stored data, it is necessary to employ information-processing technique(s) to extract the embedded information. The information engine can be regarded as an analytical toolset that consists of various analytical tools and functions to create service content through data processing, which leads to exposure of implicit and explicit information, and unexpected patterns and relationships within the data set. The employment of analytical techniques in the design

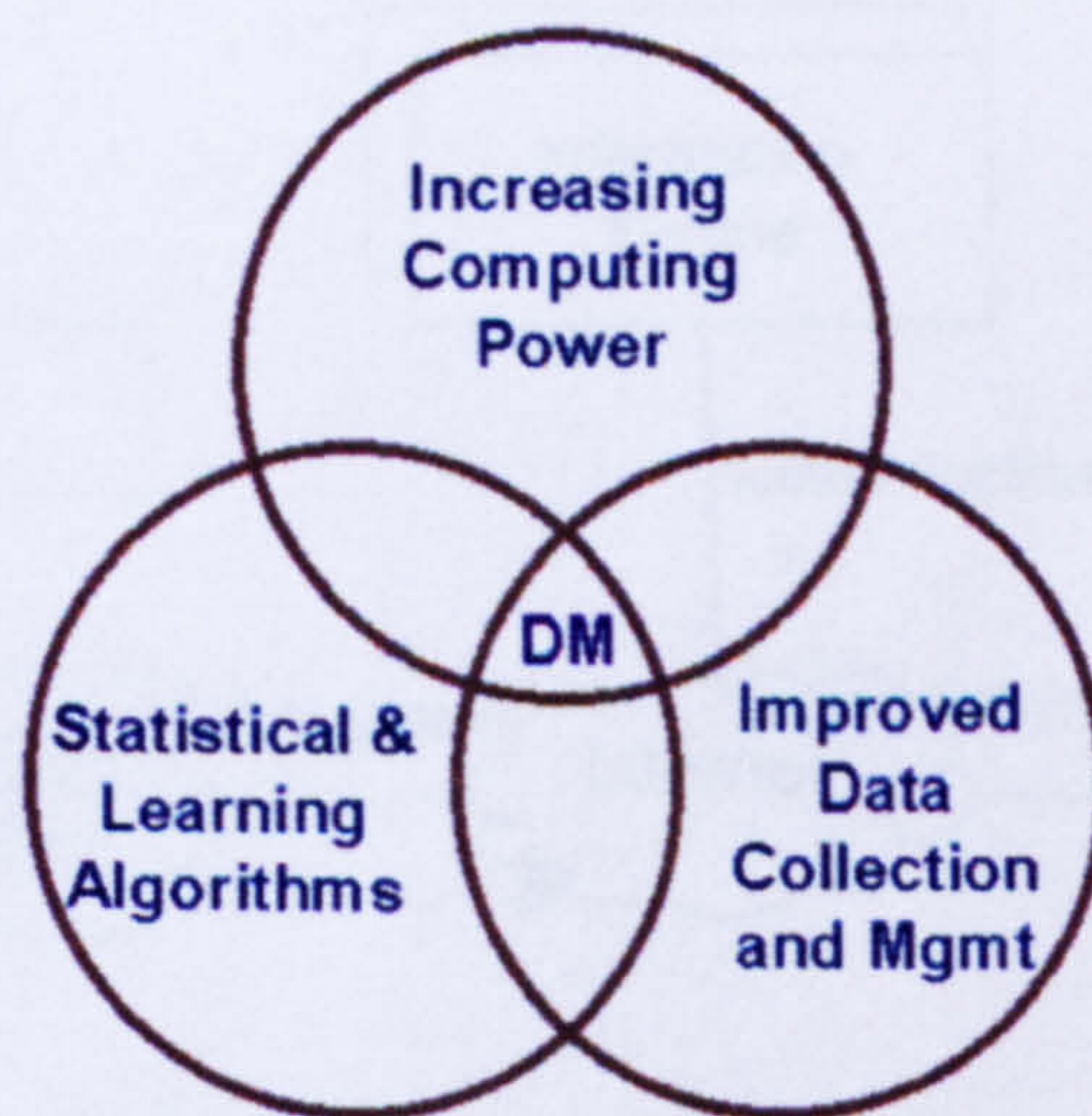


and implementation of a specific service enabler is very much dependent upon the nature of the data and their product domain(s). Figure 5.7 shows methodologies which can be adopted by information engine to create the service content.

- Statistical analysis

The traditional statistical data analysis technique is often adopted by information engine. For example, traditional statistical techniques of linear regression and nonlinear regression are often used to develop prediction model. As these techniques are well-established, they are relatively easy to use and understand (Connolly et al, 2002). Linear regression attempts to fit a straight line through a plot of data, so that the line can represent the average of all observations at that point in the plot. The problem with linear regression is that the technique only works well with linear data and is sensitive to the presence of outliers (i.e. data values do not conform to the expected norm) (Connolly et al, 2002). Nonlinear regression can avoid the main problems of linear regression; it is still not flexible enough to handle all possible shapes of the data plot.

- Data Mining (DM)



*Figure 5.8 Data mining (Thearling, 2004)*

Data mining in some way is an extension of statistics combined with some artificial intelligence and machine learning techniques (Thearling, 2004). It is a convergence of three technologies as shown in Figure 5.8. Data mining is the process of extracting valid, previously unknown, comprehensible, and actionable information from large



dataset and using it to make crucial business decisions (Han et al, 2001; Pyle, 2003). A good example of this technique used in the service enabler, would be the application of pattern analysis based on historical data.

The main limitations of the data mining include (Connolly & Begg, 2002):

- Potentially high cost
  - Identification of suitable data to mine is a challenge. Data mining requires a single, separate, clean, integrated, and self-consistent source of data. Usually a data warehouse is used for providing data for mining.
  - Data mining is still a relatively new technology.
- Consuming web services

As reviewed in the Chapter 3, Web service is a collection of functions that packaged as a single entity and published to the network for use by other programs (Glass, 2000). In general, Web services support communications among a variety of software applications, platforms and frameworks to ensure the inter-operability.

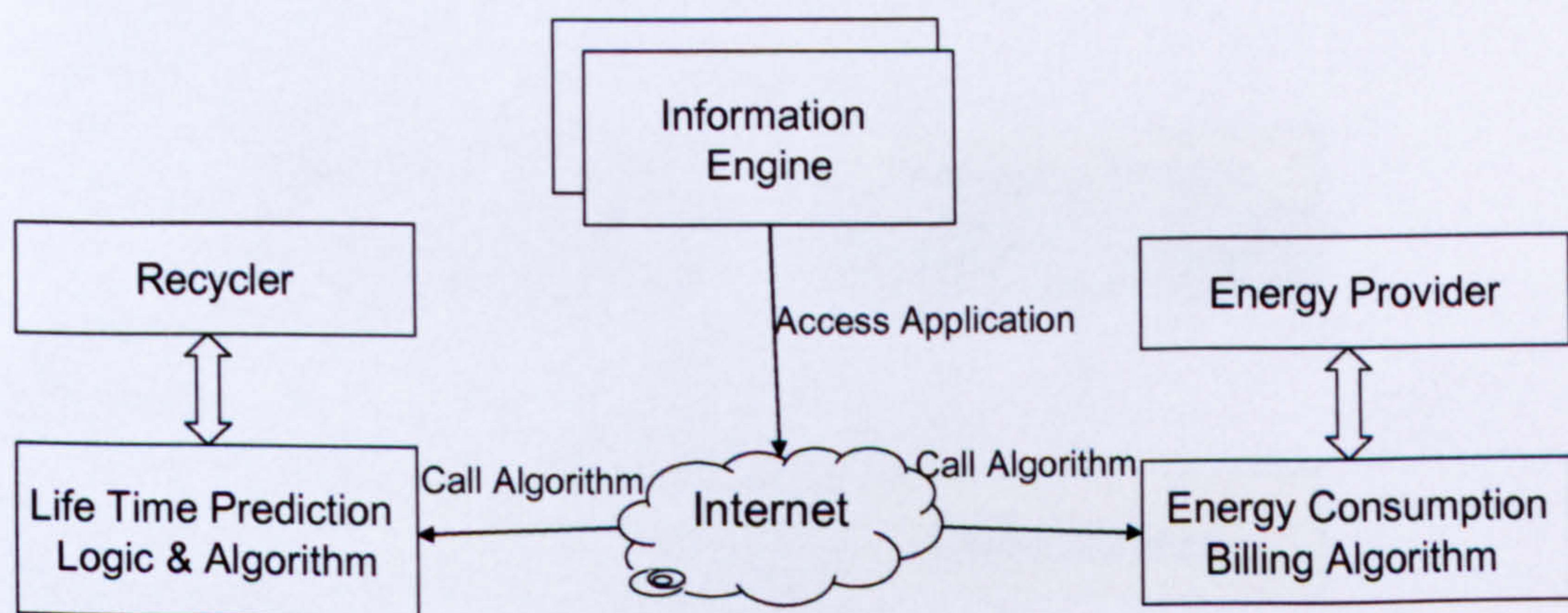


Figure 5.9 Consuming Web services

In order to support the use of third party's sophisticated and domain-specific statistical analysis functions which may reside on a variety of platforms and/or/frameworks over the Internet, *consuming web services* is another potential method which could be used to create service content by *information engine*. For example, recyclers may have some expertise algorithms and logic for predicting the remaining life of the parts at the end-



of-life product; energy providers may have specific energy consumption billing algorithm, etc. These algorithms and logic can be packaged into Web services and published to the Internet so that the *information engine* can use these exposed expertise/specific algorithms/logic to process the acquired life cycle data e.g. predicting remaining life time span , calculating energy consumption bill, as shown in Figure 5.9.

#### 5.5.2.6 Knowledge Base and Inference Engine

The knowledge base normally contains both factual and heuristic knowledge (Feigenbaum et al, 1993). Factual knowledge is a knowledge that is widely shared, typically found in textbooks or journals, and commonly agreed upon by those knowledgeable persons in the particular field; while heuristic knowledge is the less rigorous, more experiential, more judgmental knowledge of performance (Feigenbaum et al, 1993). In contrast to factual knowledge, heuristic knowledge is rarely discussed, and is largely individualistic. It is the knowledge of good practice, good judgment, and represents the knowledge compiled by an expert through the experience of solving past problems (Durkin, 1994; Feigenbaum et al, 1993). It is the knowledge that underlies the "art of good guessing" (Feigenbaum et al, 1993).

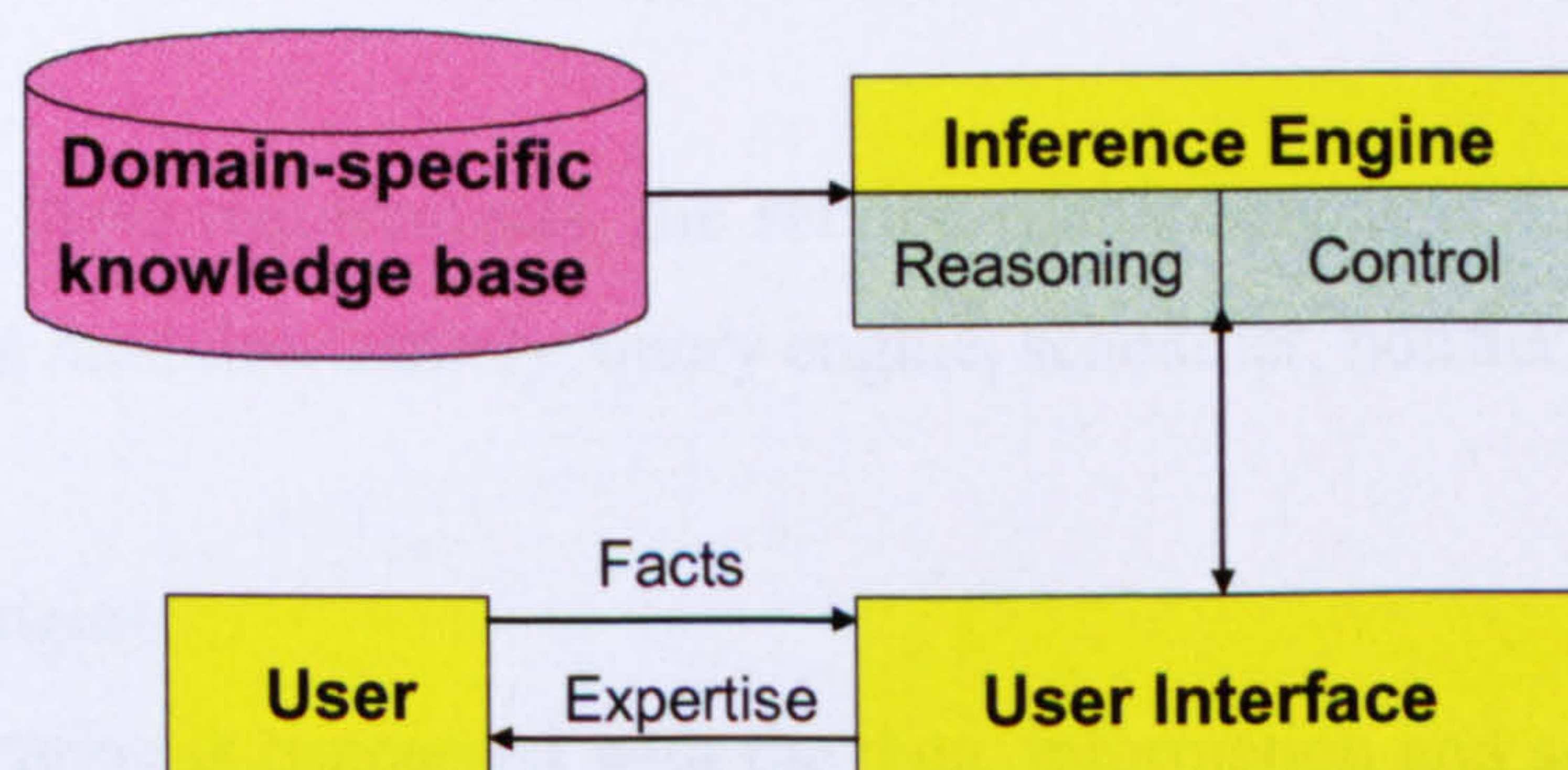


Figure 5.10 Expert system (Sell, 1985; Klein et al, 1990; Feigenbaum et al, 1993; Durkin, 1994)

The knowledge base in service enabler contains the product domain specific knowledge from which the inference engine draws conclusions. These conclusions are responses to the stakeholders' queries for advice. The knowledge set in the knowledge base are normally product related expertise and knowledge, which are generally provided by the manufacturers. Service providers or stakeholders supply facts or other information to



the knowledge base and receive advice in response. The inference engine provides reasoning mechanism used to infer the new knowledge from the existing facts. Two fundamental reasoning approaches are forward chaining (i.e. data driven) and backward chaining (i.e. hypothesis driven) techniques (Klein et al, 1990). Essentially an expert system is created by the combination of the domain-specific knowledge with the inference engine which can process the knowledge in response to a user's request for suitable advice, as shown in Figure 5.10 (Sell, 1985; Klein et al, 1990; Feigenbaum et al, 1993; Durkin, 1994)

#### 5.5.2.7 Service Enabler Manager

The service enabler manager is the kernel of the service enabler that administers the data/information flow between various internal modules and external actors. Hence it should provide the following functionalities and services:

- Data/information/knowledge/service content query service
- Scheduling the pre-defined tasks
- Notification service
- Transformation of data/information/knowledge/service content into the appropriate format

Considering these requirements, the service enabler manager is designed to consist of the following modules, namely, query engine, scheduler, notifier and report generator.

- Query Engine

The query engine is concerned with the data, information and service content retrieval. It is also responsible for raising the request to information engine to perform necessary information processing tasks. Generally it should support query by product ID, dynamic data name, date & time, etc.

- Scheduler

An application component that allows users to schedule regular mandatory operations, for example, performs statistical analysis on the IDU data to ensure abnormality can be



detected. Appropriate pre-defined action(s) could be triggered automatically through the scheduler, for example, raise a request to notifier to alert the appropriate actor(s).

- Notifier

The notifier is responsible for notifying the appropriate actor(s) via the pre-defined mechanism (e.g. email, SMS) when pre-defined condition is met. The notification message could be alert message or advice on certain events.

- Report Generator

The report generator is designed to present the requested data, information or service content in a structured presentation, for examples, charts and document in a pre-defined format.

## **5.6 Relationship between the Proposed Methodology and ELIMA Project**

The reason why this research study was carried out in association with ELIMA project has been discussed in the literature survey (Section 2.5.3). In this section, we further illustrate the relationship between the proposed PSS realisation methodology and ELIMA project as follows (Figure 5.11):

- IMS vs. ELIMA project

Two IMSs are one of the main deliverables in ELIMA project developed to support ELIMA white goods field trial (i.e. fridge/freezer) and brown goods field trial (i.e. game console). They are used to house and manage life cycle data for fridge/freezer and game console generated during field trials in an integrated manner.

- Service enabler vs. IMS

The service enabling scheme is proposed initially inspired by ELIMA project. Hence according to the definition of service enabler, two IMSs in ELIMA are actual instances of the service enabler. In other words, service enabler is an abstraction of IMSs.



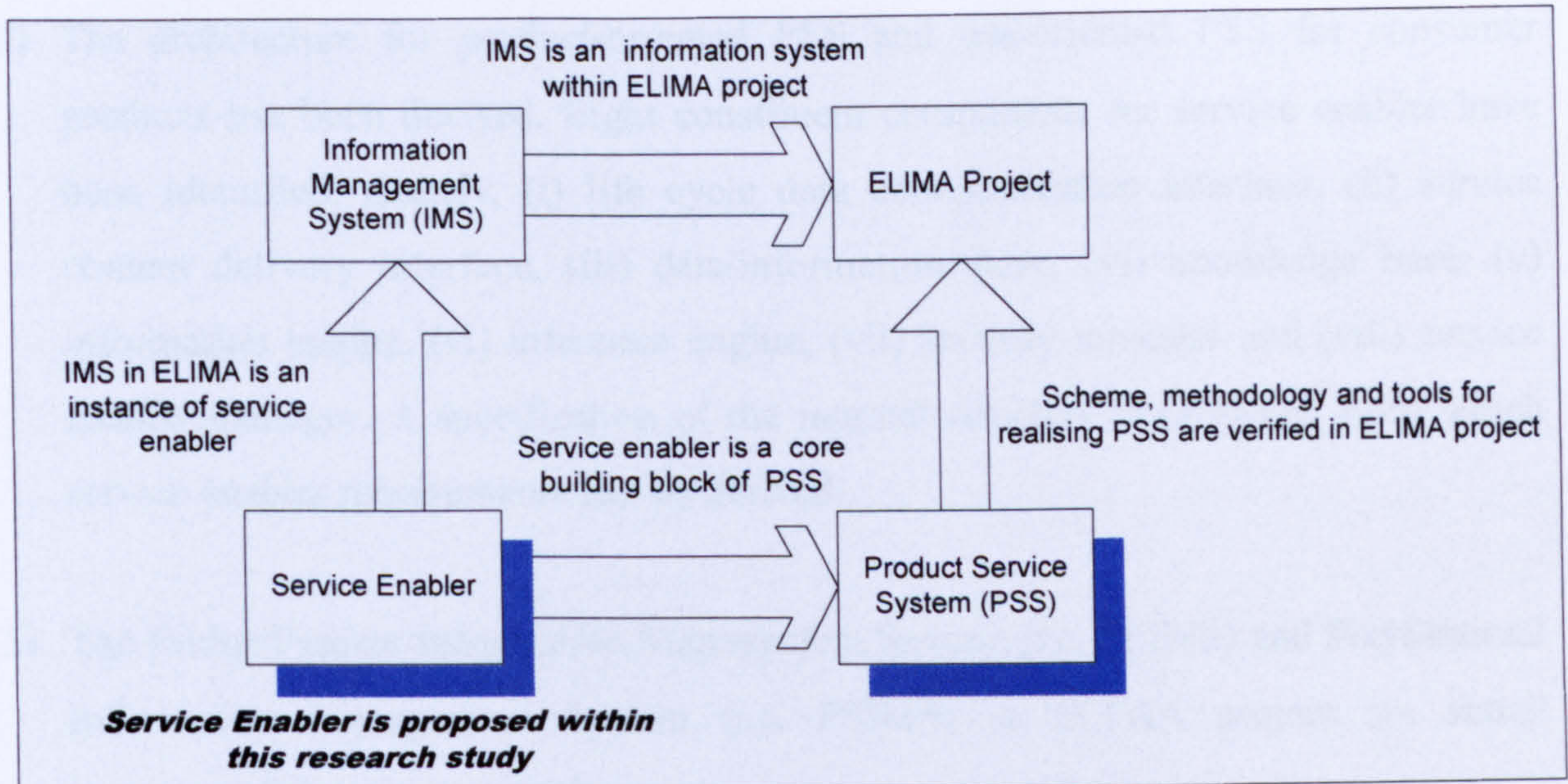


Figure 5.11 Relationship between research study and ELIMA project

- Service enabler vs. PSS

Service enabler is a core building block within PSS.

- ELIMA project vs. PSS

Scheme (i.e. service enabling scheme), methodology (i.e. PSS realisation methodology) and tool (i.e. the Framework) for realising PSS for consumer products will be tested in ELIMA project through two product-oriented PSS prototypes developed in ELIMA field trial.

## 5.7 Summary

- 1) A conceptual model and methodology for realising product-oriented PSS and use-oriented PSS for consumer products have been proposed based on the service enabling scheme. In the methodology, the realisation of PSS is through the following 3 steps:
  - (i) Design appropriate IDU to acquire product life cycle data.
  - (ii) Transmit life cycle data to service enabler.
  - (iii) Develop a service enabler. The research study concentrates on the design and development of service enabler.



- 2) The architecture for product-oriented PSS and use-oriented PSS for consumer products has been devised. Eight constituent components for service enabler have been identified, namely, (i) life cycle data communication interface, (ii) service content delivery interface, (iii) data/information store, (iv) knowledge base, (v) information engine, (vi) inference engine, (vii) security manager and (viii) service enabler manager. A specification of the internal structure is produced from which service enabler requirements can be derived.
  
- 3) The Fridge/Freezer Information Management System (i.e. FFIMS) and PlayStation2 Information Management System (i.e. PSIMS) in ELIMA project are actual instances of the service enabler.



## Chapter 6 Analysis and Design of the Framework

### 6.1 Introduction

The development of service enabler has been identified as a key and challenging procedure during the realisation process of PSS for consumer products. In order to facilitate the development of service enabler, an associated software toolkit, namely, the Framework is developed. This Framework is distinct from the PSS architecture defined previously. It is a software framework which comprises a run time infrastructure, schemas, class libraries/APIs and components aiming to facilitate the practical creation and implementation of product-specific service enabler. In addition, it also means a typical primitive service enabler skeleton.

As part of this research, the rest of the thesis will describe analysis, design and implementation of the Framework, and test procedures undertaken. The key stages in this process together with associate activities are as follows:

- A. Analysis of two IMSs in ELIMA to generate design specification for the design of the Framework.
- B. Architectural design to develop a 4-layer Framework model
- C. Detailed design of individual layers in the Framework.
- D. Implementation of the Framework
- E. Description of how to use the Framework to create product-specific service enabler
- F. Using the Framework to create a fridge/freezer service enabler (i.e. FFIMS) and PS2 service enabler (i.e. PSIMS), and integrating FFIMS and PSIMS into ELIMA field trials to examine the proposed PSS realisation methodology and associated toolkit.

This is shown in the Figure 6.1 A, B, C, D, E and F:



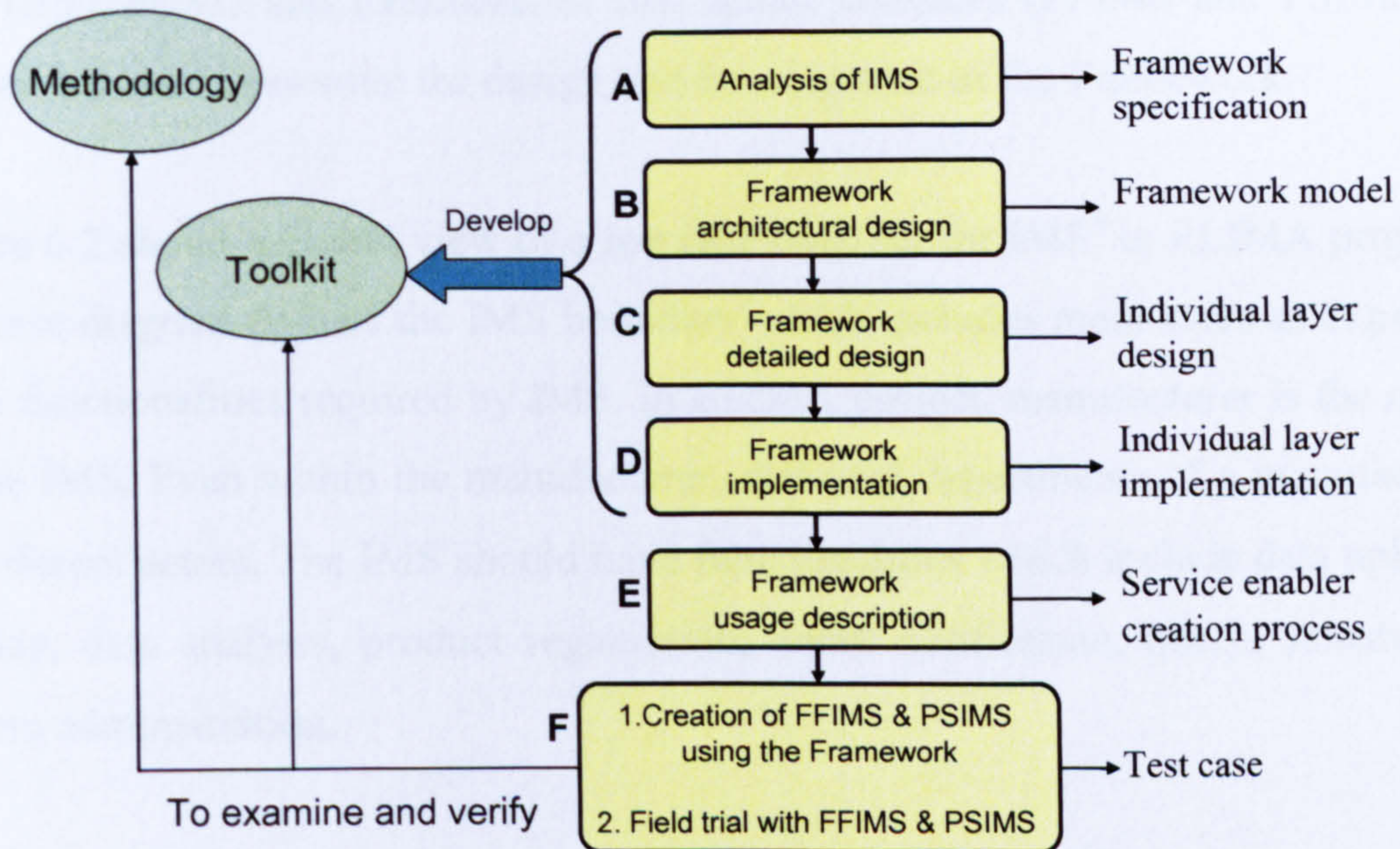


Figure 6.1 Development and test plan

In the initial design of the Framework, we may regard the two IMSs in ELIMA project as typical instances of a service enabler. Two IMSs in conjunction with ELIMA field trials can provide a broad specification for the Framework design. Hence the Framework requirements specification is derived from the IMS use case analysis and IMS three-tier distributed architecture.

As defined previously that “service enabler can be an intelligent information management system or expert system or combination of both“, this allows us to proceed with the development of the Framework without implementing *knowledge base* and *inference engine*, as these two elements can be very approximately be replaced by *data/information store* and *information engine*. These can provide sufficient test capability for the purpose of testing the operation of the Framework without undue impact on the general functionality of the service enabler.

## 6.2 IMS Use Case Global View

An excellent technique to improve understanding of the requirements is the creation of use cases narrative of description of domain process (Larman, 1998). Understanding the



user requirements and intentions of two actual instances (FFIMS and PSIMS) of the service enabler is essential for design and development of the Framework.

Figure 6.2 shows a global view of a use case diagram for IMS<sup>3</sup> in ELIMA project. This use case diagram defines the IMS boundary which includes main users as expected and main functionalities required by IMS. In ELIMA project, manufacturer is the main user of the IMS. Even within the manufacturer, different departments of a manufacturer act as different actors. The IMS should have functionalities which include data upload, data display, data analysis, product registration, email notification, query, scheduling and system administration.

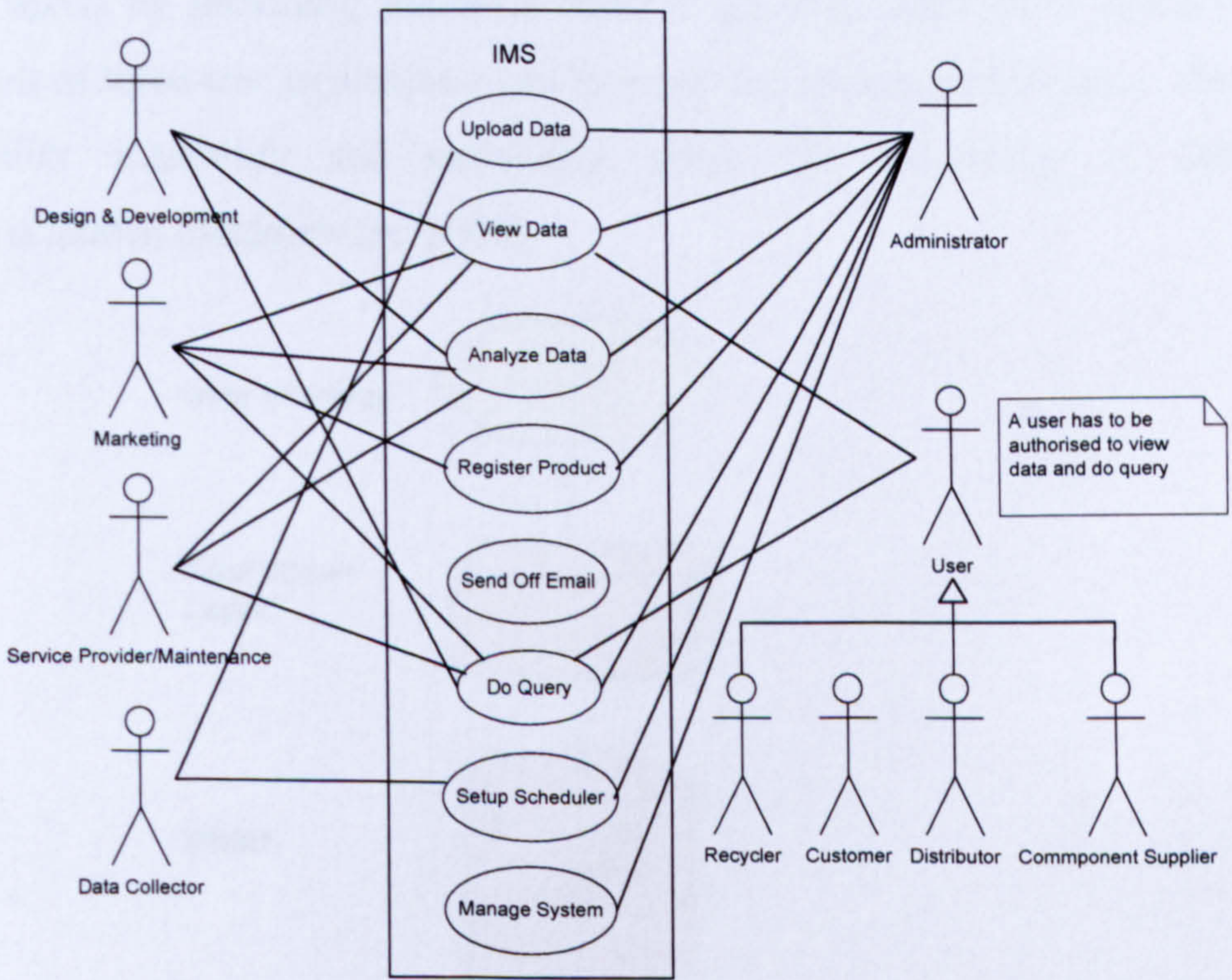


Figure 6.2 Use case global view of IMS in ELIMA

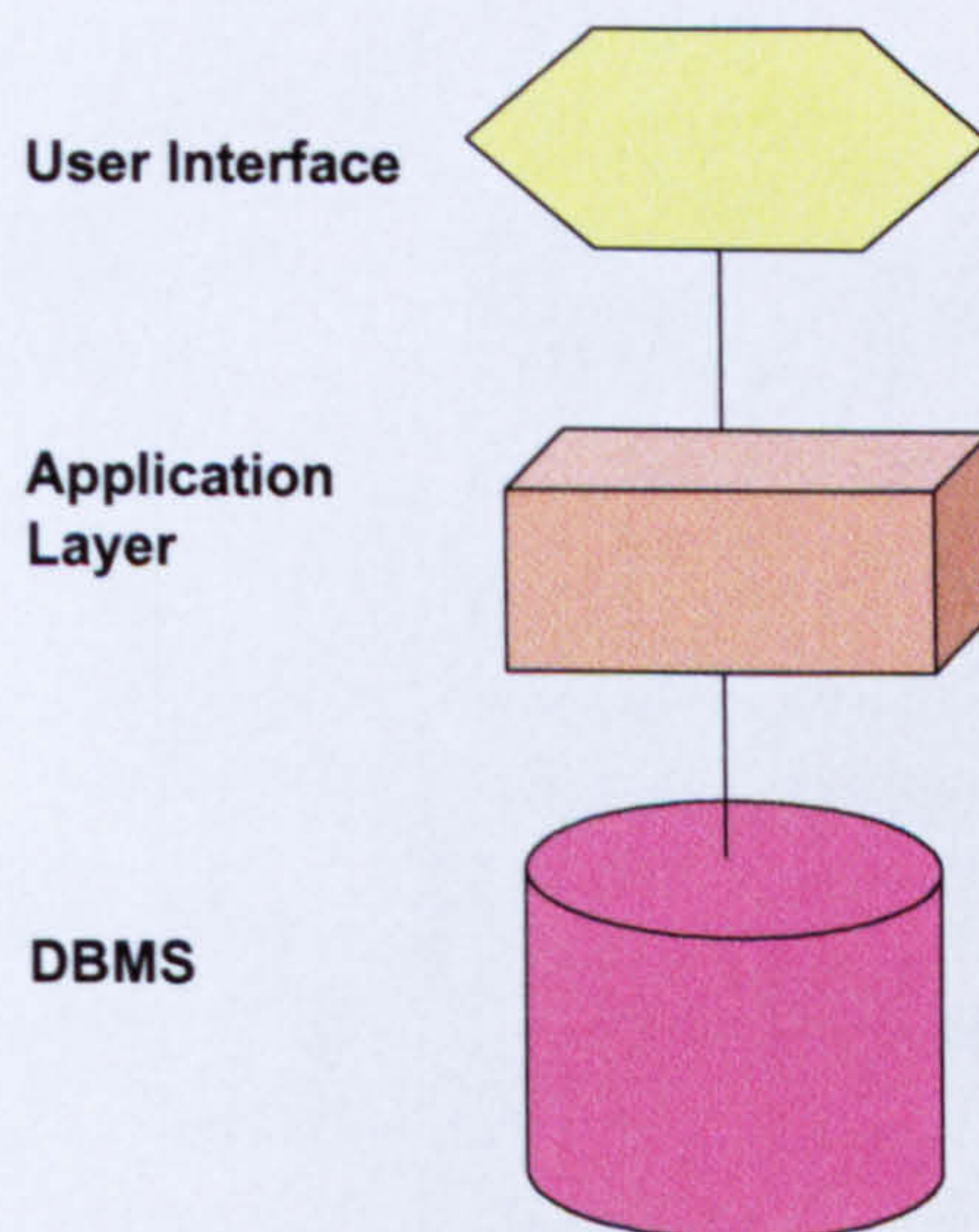
<sup>3</sup> IMS here refers two IMSs in ELIMA, i.e. FFIMS and PSIMS



## 6.3 IMS Three-tier Distributed Architecture

### 6.3.1 Information System Three-tier Architecture

In order to overcome the limitations of two tier client/server architecture (i.e. interface and application) such as the performance deteriorates when the number of user exceeds 100, the information system three-tier distributed architecture emerged in the 1990s (Middleware, 2004). The third tier (middle tier server) is between the user interface (client) and the data management (server) components, as shown in Figure 6.3. This middle tier provides process management where business logic and rules are executed and can accommodate hundreds of users (the two tier architecture can only support about 100 users) by providing functions such as queuing, application execution, etc. The adoption of three-tier architecture can increase the system performance, flexibility, maintainability reusability and scalability, while the complexity of distributed processing is hidden (Middleware, 2004).



*Figure 6.3 Three-tier distributed architecture (Middleware, 2004)*

### 6.3.2 Three-tier Architecture of IMS in ELIMA

The three-tier distributed architecture adopted in the IMSs and how various parts work together are illustrated in Figure 6.4.



- *External world* - is concerned with user interface and user functions such as life cycle data acquisition, service content delivery, text input, Java Applet, dialog, and display management, etc. It also consists of various service enabler actors such as IDU, manufacturer and recycler etc.
- *Life cycle information management* - the core business logic (i.e. data analysis) is residing in this middle tier. It is responsible for receiving product life cycle data, processing the data and providing capabilities to obtain information, discover knowledge, etc. It acts as a bridge between the users and the back-end database.
- *DBMS* - this layer consists of a data store with a range of management components including data manipulation, security monitor, transaction management, concurrency control and database administration. This layer provides persistent data storage and database management functions to house product life cycle data and information.

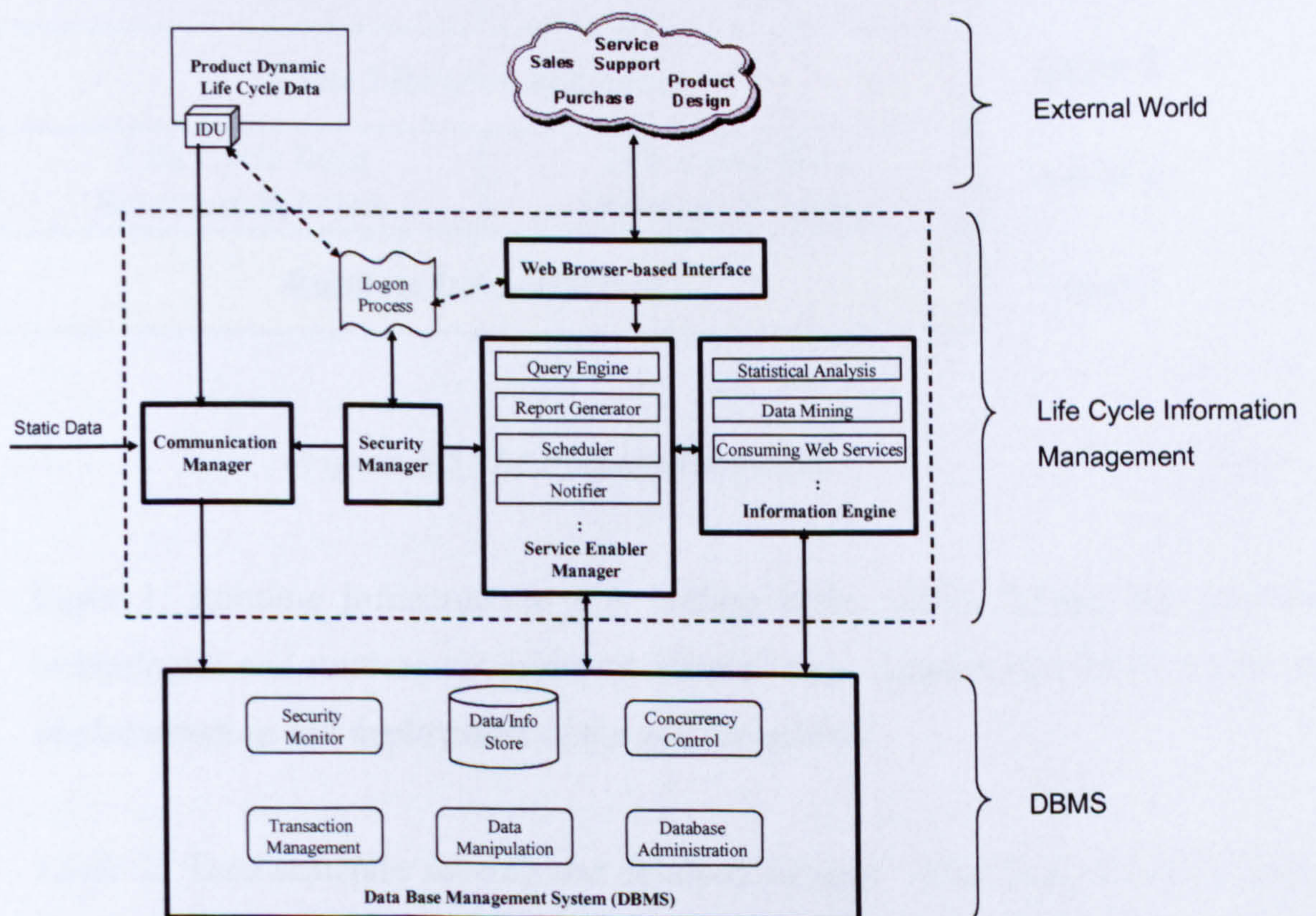


Figure 6.4 IMS three-tier distributed architecture



6.4 The Framework Architectural Design

Design is the process of specifying a solution that is consistent with the requirements specification (Douglass, 1999). The design can be categorised into two primary categories: architectural design and detailed design. The architectural design details the large-scale software structures, such as subsystems, packages, and tasks; while detailed design specifies the internal primitive data structure and algorithms (Douglass, 1999). The architectural design of the Framework focus on the development of software framework model; while the detailed design includes life cycle data design, database design, class libraries/APIs design and a range of functional components design (e.g. security manager, communication manager, etc).

As a result of architectural design, the Framework model is devised. This Framework model is layered into four, as shown in Figure 6.5.

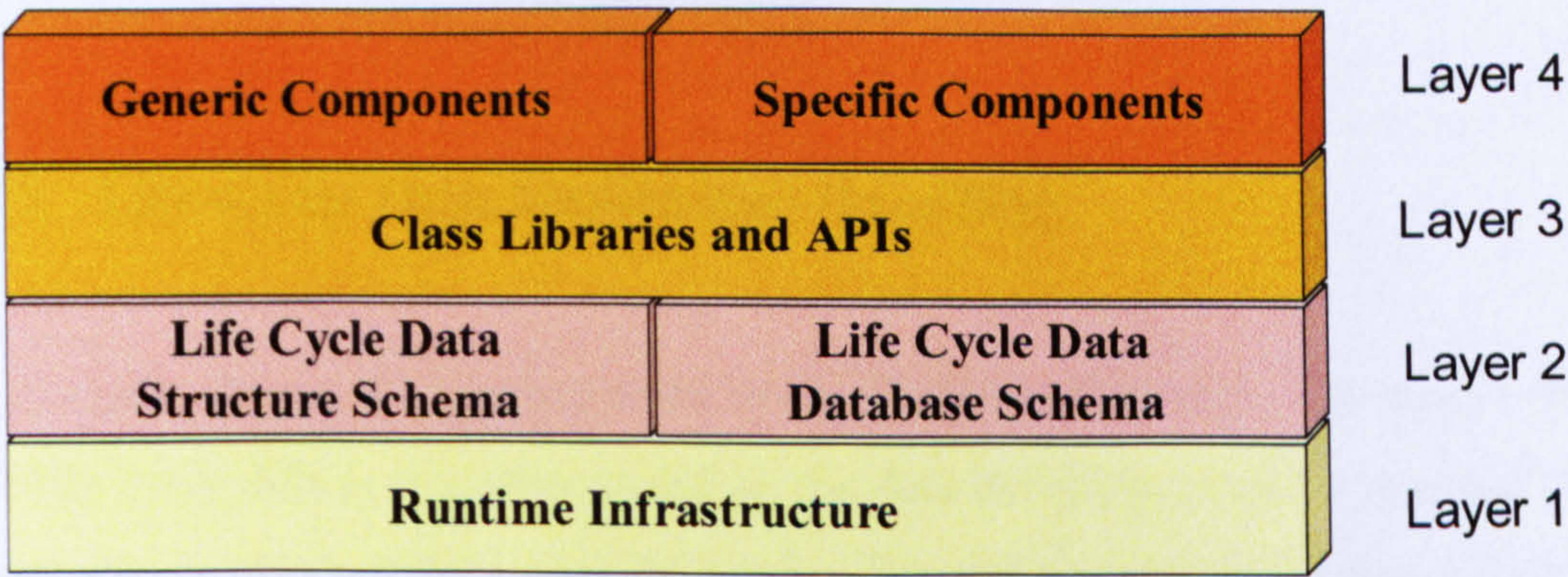


Figure 6.5 The Framework model

- Layer 1: Runtime infrastructure - A bottom layer, which defines the platform, technologies and runtime environment adopted (e.g. Apache-PHP-MySQL) for the implementation and deployment of the service enabler.
- Layer 2: Data structure schema and database schema - This layer defines dynamic data structure schema and database schema. Data structure schema is mainly concerned with the definition of an appropriate data structure which is both machine and human readable. It is used for verification, presentation, communication and



process of the dynamic life cycle data. Database schema is to provide an overall description of the life cycle database for service enabler. The database schema is DBMS independent, means it is regardless of whatever DBMS (e.g. MySQL, SQL Server) being chosen.

- Layer 3: Class libraries and APIs - This layer provides class libraries and APIs to support the development of product-specific service enabler and the development of top layer components.
- Layer 4: Components - This top layer defines software components within the service enabler. The component is a unit of executable deployment that plays a part in a component kit (Graham et al, 2001). The components are classified into generic components and product-specific components in this research study. The development of components employs class libraries and APIs provided in third layer.

## 6.5 Life Cycle Data Design and Handling

Life cycle data is the starting point of the Framework design. The design and handling of life cycle data is challenging due to the data diversity, even for a single product. For example, in terms of data acquisition, the data can be categorised into dynamic data and static data (Scheidt & Zong, 1994); in terms of dynamic data processing, the data can be categorised into *primary dataset* and *extended dataset*. Furthermore, there are few general rules which can be referenced for life cycle data design and handling. Hence, a general life cycle data handling guideline for consumer products is presented.

Since the dynamic data varies throughout the product life, while the static data rarely changes, the handling processes for static data and dynamic are different.



### 6.5.1 Static Data Handling

The static data generally originates from the manufacturer and take the form of Excel spreadsheet (e.g. BoM), specification (e.g. end-of-life treatment specification), XML document or database. It is particularly important for manufacturers and recyclers at the product end-of-life treatment in compliance with WEEE and RoHS directives. In order to accommodate this need, a three - level static data handling guideline is proposed and illustrated as follows (Figure 6.6).

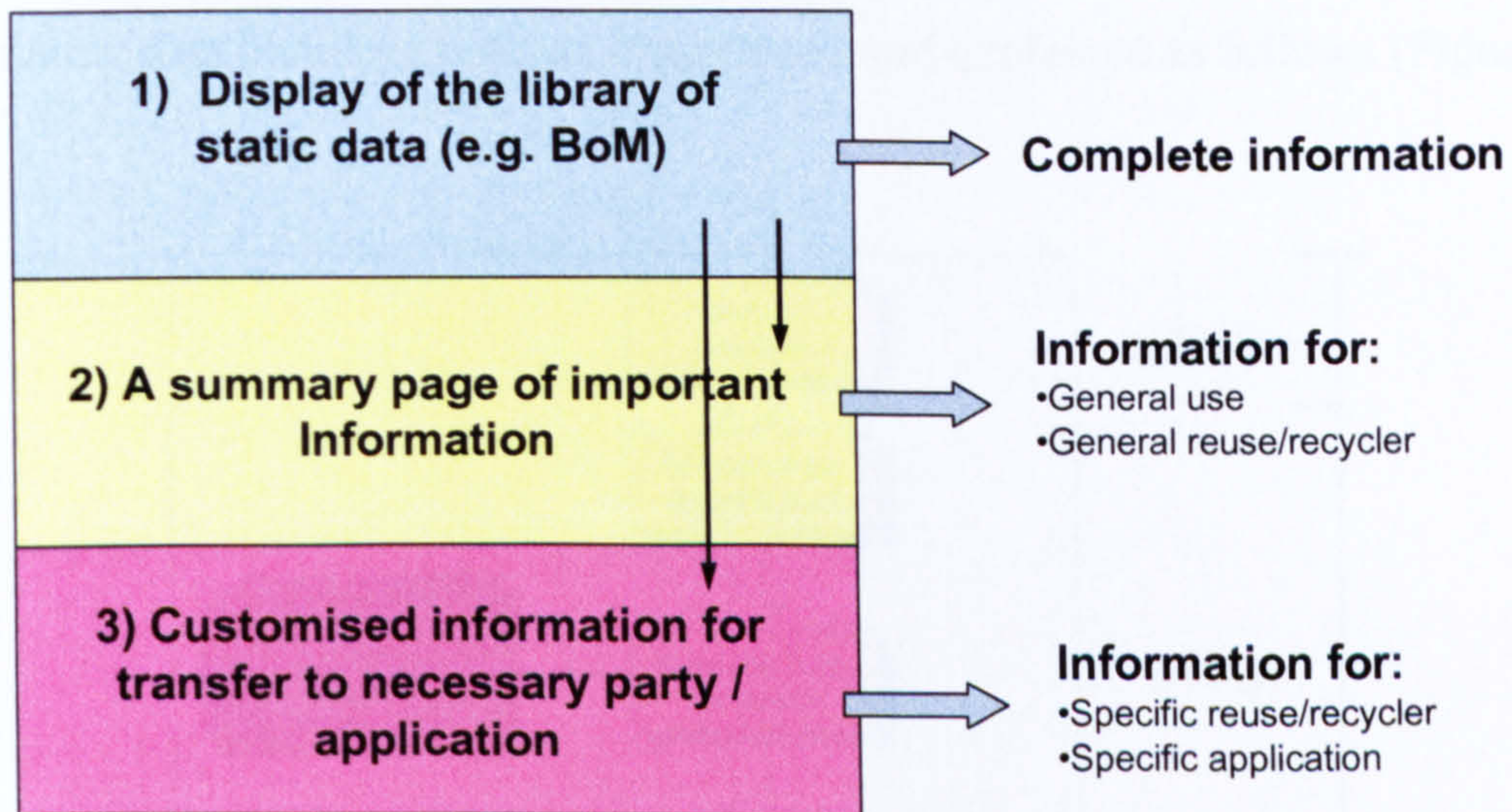


Figure 6.6 A general static data handling

- 1) In the first level, display of the library of static data. As usually the static data takes the form such as Excel spreadsheet, specification, etc, display of the static data can provide valuable information for the suitable services to be delivered. For example, the display of BoM of different product model and the display of end-of-life treatment specification can facilitate the delivery of end-of-life treatment service.
- 2) In the second level, a summary page of important information related the product is provided as general information that can help manage the end-of-life product. For example, the summary information from BoMs such as *major components for reuse, hazardous materials, material composition in percentage*, etc. In addition to this, it is useful to combine some key findings from the collected dynamic data such as *the remaining life of key components based on their usages with necessary remarks*, etc.



- 3) In the third level, the service enabler should supply customised information about the static data in a *de facto* standard electronic format (i.e. XML) for presentation, communication and exchange. This information presented in XML could be sent to third party's application, e.g. an application in reuse / recycler that has specialised algorithm for predicting the remaining life span.

### 6.5.2 Dynamic Data Handling

The dynamic data handling process is proposed and explained as follows (Figure 6.7):

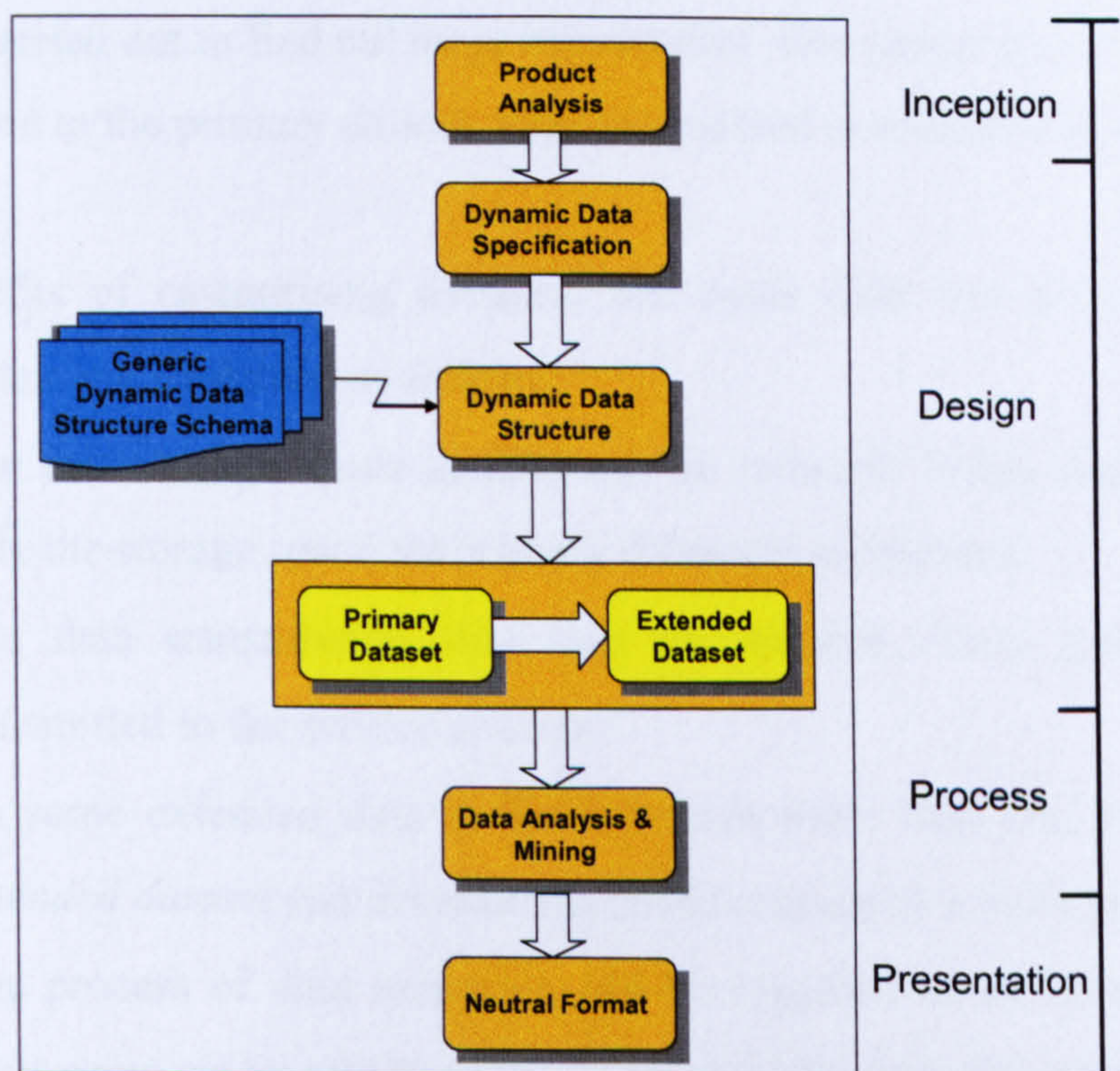


Figure 6.7 General dynamic data handling

- 1) Analyse product to generate a data specification. The data specification is an important piece of information that describes the data to be collected from the product. It includes the information such as the dynamic data item to be recorded (e.g. temperature, power-off events), how the data is recorded (i.e. time-based logging, event-based logging), etc,



- 2) Derive a data structure from the *generic data structure schema*, which is developed in this research study to help with the generation of the dynamic data structure. Normally the data structure is presented in XML schema or DTD.
- 3) Define the *primary dataset* and *extended dataset*. As constraints of IDU storage space, data transmission time, etc, not all dynamic data can be collected and stored at IDU. On the other hand, more data is often required to accurately describe the product usage, monitor product or enable better services. In order to address these limitations, we categorise dynamic data into *primary dataset* and *extended dataset*. In principle an original dataset directly collected from the product is defined as *primary dataset*. Based on the acquired *primary dataset*, initial data processing could be carried out to find out more information. The results are often referred to as an extension to the primary dataset, therefore named as *extended dataset*.

The benefits of categorising dynamic life cycle data into primary dataset and extended data are explained as follows:

- The data storage space at IDU can be reduced. When designing the IDU, only the storage space for primary dataset is considered.
  - The data transmission time can be reduced. Only primary dataset is transmitted to the service enabler.
  - As some extended data is derived from more than one primary data, the *extended dataset* can describe the product usage in a more precise manner.
  - The process of data extension can be regarded as an initial data analysis. Thus some useful information can be obtained from the *extended dataset*.
- 4) Carry out data analysis/data mining on both *primary dataset* and *extended dataset*.
  - 5) Present the processed result in a neutral format (e.g. XML) so that the result can be consumed by other business (e.g. recycler) regardless of whatever platform or operating system is used.



### 6.5.3 Development of Generic Life Cycle Data Structure Schema

The design of data structure in dynamic data handling process is a challenging and important task. The data structure varies from product to product. Furthermore, the presentation of dynamic data should follow a certain pre-defined format or standard so that the data processing algorithm or logic provided by the Framework can be applied. In order to address this issue, we developed a *generic dynamic data structure schema* to facilitate the design of dynamic data structure.

#### 6.5.3.1 Dynamic Data Acquisition Modes

It is helpful to overview the data acquisition modes before generating a data structure schema. In general, we can classify the mode of dynamic data collection into two modes: (i) *statistical* mode, i.e. data to be collected is an accumulative value up to that collecting time point. It can be sum of events (totals / counts), sum of operating time (total duration / time), etc (ii) *data logging* mode, i.e. data is logged every fixed time interval or when an event happens. *Data logging* can be categorised into three types: time-based data logging, event-based data logging and message-based data logging. The description of these data logging methods and main features are described as below:

- *Time-based logging* refers to logging the instantaneous value of the specified data field (e.g. energy, temperature) at a specified time interval (e.g. every minute, every five minutes, every hour).
- *Event-based logging* means logging the details of the specified data field (e.g. event name, value, happening time) only when certain event happens (e.g. over temperature event, fast speed event, door opening and closing).
- *Message-based logging* normally means manually logging messages such as error/failure, maintenance carried out, and part replaced, etc.



These two data collection modes (i.e. statistical and data logging) are all adopted in ELIMA field trials. Figure 6.8 is a high level view of dynamic data structure derived with reference to dynamic data acquisition modes to support ELIMA field trials.

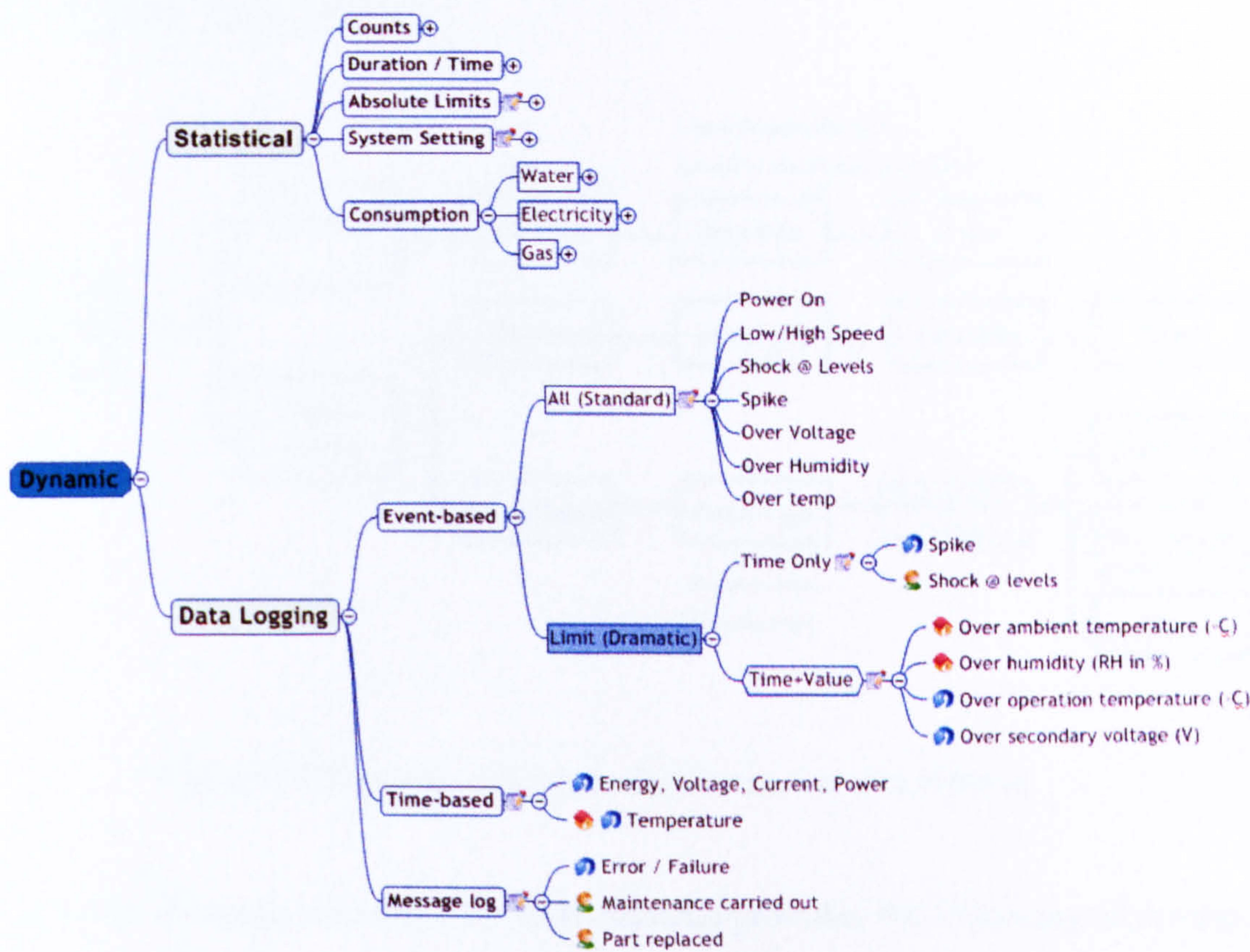


Figure 6.8 Dynamic data structure in field trials

6.5.3.2 Generic Dynamic Data Structure Schema

In order to facilitate understanding and development of generic dynamic data structure schema for consumer products, we develop a description of dynamic data structure in the first instance with reference to dynamic data acquisition modes, as shown in Figure 6.9.

Actually the produced data structure description can be used as a reference for defining the dynamic life cycle data structure. However, in most cases, the life cycle data is presented in XML document. The data structure description presented in this normal way may not well support presenting data in XML document. Furthermore, it can not be



used by applications to check the structure validity of the data presented in XML document. In order to tackle these limitations, a generic dynamic data structure schema is developed by incorporating XML schema technology, which is a formalisation of the constraints expressed as rules or a model of structure applied to XML document (Connolly, 2002; Vlist, 2002).

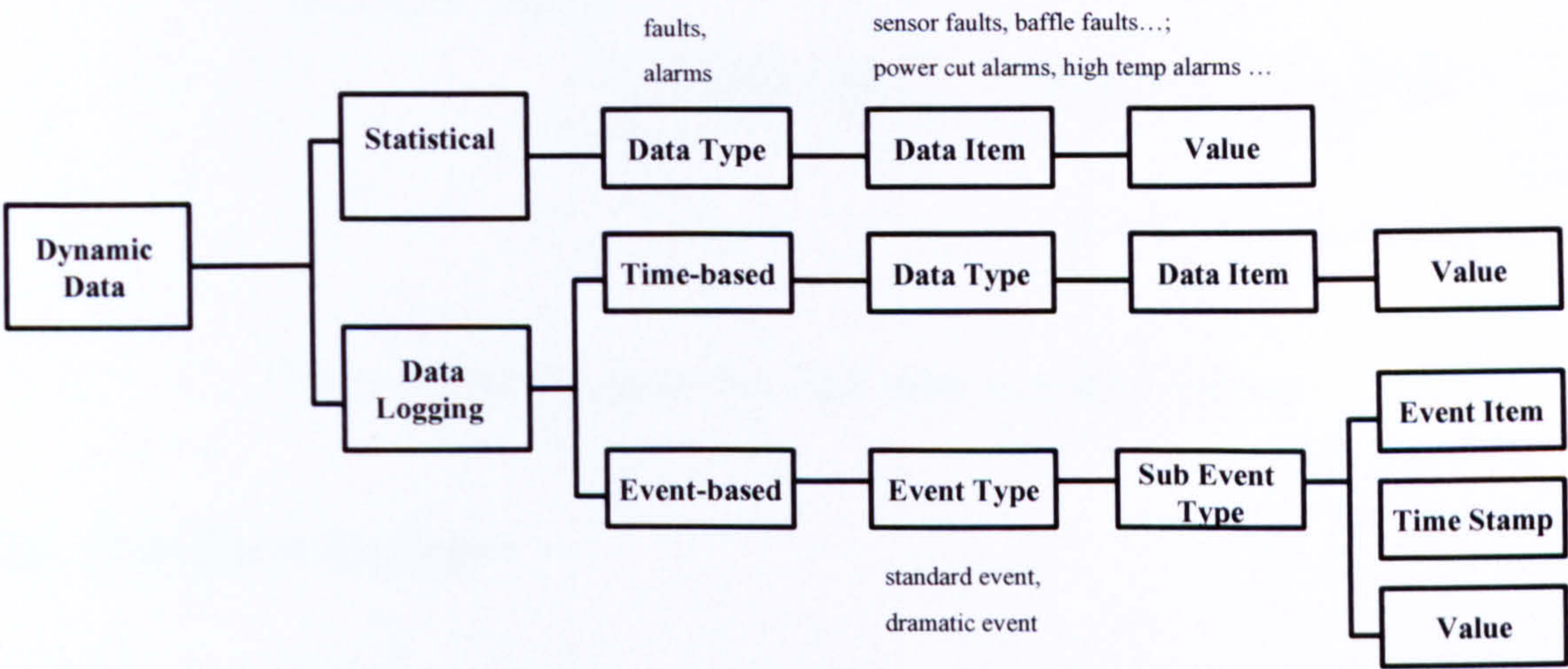


Figure 6.9 Dynamic life cycle data structure description

The dynamic structure presented by XML schema provides the following advantages:

- XML schema has support for data types. This allows description of permissible document content, validation of data correctness, easier working with data from a database, easier definition of data facets (restrictions on data), easier definition of data patterns (data formats), and easier conversion of data between different data types.
- XML schema adopts XML syntax. The experience and tools from XML could be used to speed up development.
- XML schema secures data communication. Sender can describe the data in a XML schema that only the receiver will understand.
- XML schema is extensible, just like XML, as they are written in XML.

Figure 6.10 shows the generic dynamic life cycle data structure schema, which can be referenced for dynamic data structure definition for a specific consumer product. The



schema is derived and developed from generic dynamic data structure description. Variation of data can be dealt with by extending the schema due to its flexibility.

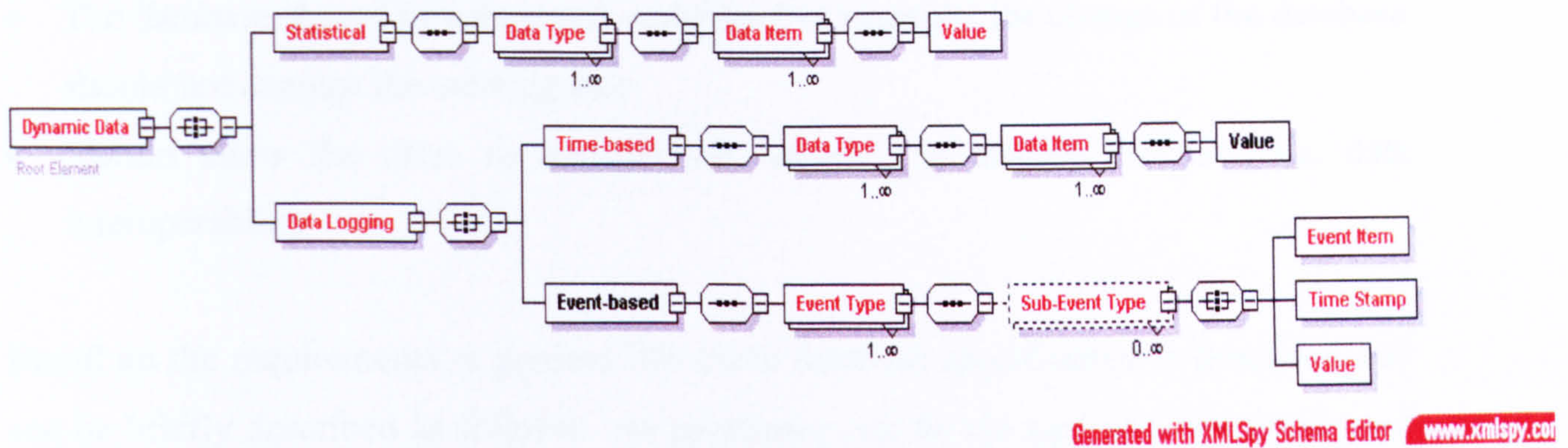


Figure 6.10 Dynamic life cycle data structure schema

## 6.6 Database Design

A database, together with the associated housekeeping functionalities (e.g. data manipulation, concurrency control), known as DBMS, is certainly a core module in the service enabler. The database not only provides the persistent storage for life cycle data (including primary dataset and extended dataset) and various stakeholders' profile, but also provides capabilities to allow different applications to share the data. Furthermore, it can ensure the data security and integrity. Thus the role of the database is vital.

### 6.6.1 Product Life Cycle Database Specification

In order to facilitate the life cycle database design, a typical product life-cycle database specification is produced from both service enabler requirements specification and ELIMA field trials specification. The product life-cycle database should:

- Integrate product life cycle data of each phase (e.g. manufacturing, usage, recycling) during a product life cycle
- Integrate data of various stakeholders (e.g. manufacturer, distributor, recycler) involved during a product life cycle
- House lifecycle data to facilitate the delivery of suitable services (e.g. remote diagnostics, remote monitoring)



- House system data such as user manual, task scheduling parameters, etc to improve service enabler performance
- The database should be robust and scalable. For example, the change of the database should not damage the existing data
- Should allow the users to transparently access and modify the data, i.e. data interoperability

Based on the requirements, a product life cycle database specification is produced and can be briefly described as follows: *An appliance can be the type of white goods and brown goods. The appliance (e.g. fridge/freezer) of a specific product model (e.g. COM70, FZR40P) is directly related to a product end user. Product end user may be required to complete a questionnaire. Product end user is directly linked to distributor and can obtain the information about maintenance and maintenance providers. The appliance is equipped with an IDU to acquire the product life cycle data. Database users can query dynamic life cycle data and static data (e.g. component, materials, and subassembly) through the appliance ID. Database users can also obtain the information (e.g. name, contact details) of various stakeholders (e.g. manufacturer, recycler, distributor, component supplier) during the product life cycle. The database supports role and session paradigm and three-level access control mechanism through the role and data group.*

### 6.6.2 E-R Modelling Diagram

As the Relational Database Management System (RDBMS) is chosen for housing the life cycle data, the Framework should be designed to support the accommodation of various relational databases (e.g. MySQL, MS SQLServer) into the service enabler. In order to achieve this, the database E-R modelling technique is employed to develop a database schema.

A life cycle database E-R modelling diagram is produced as shown in Figure 6.11. This E-R model is a logical representation of life cycle database in a higher level. It can be



converted into a database schema for implementation regardless of whatever DBMS being used.

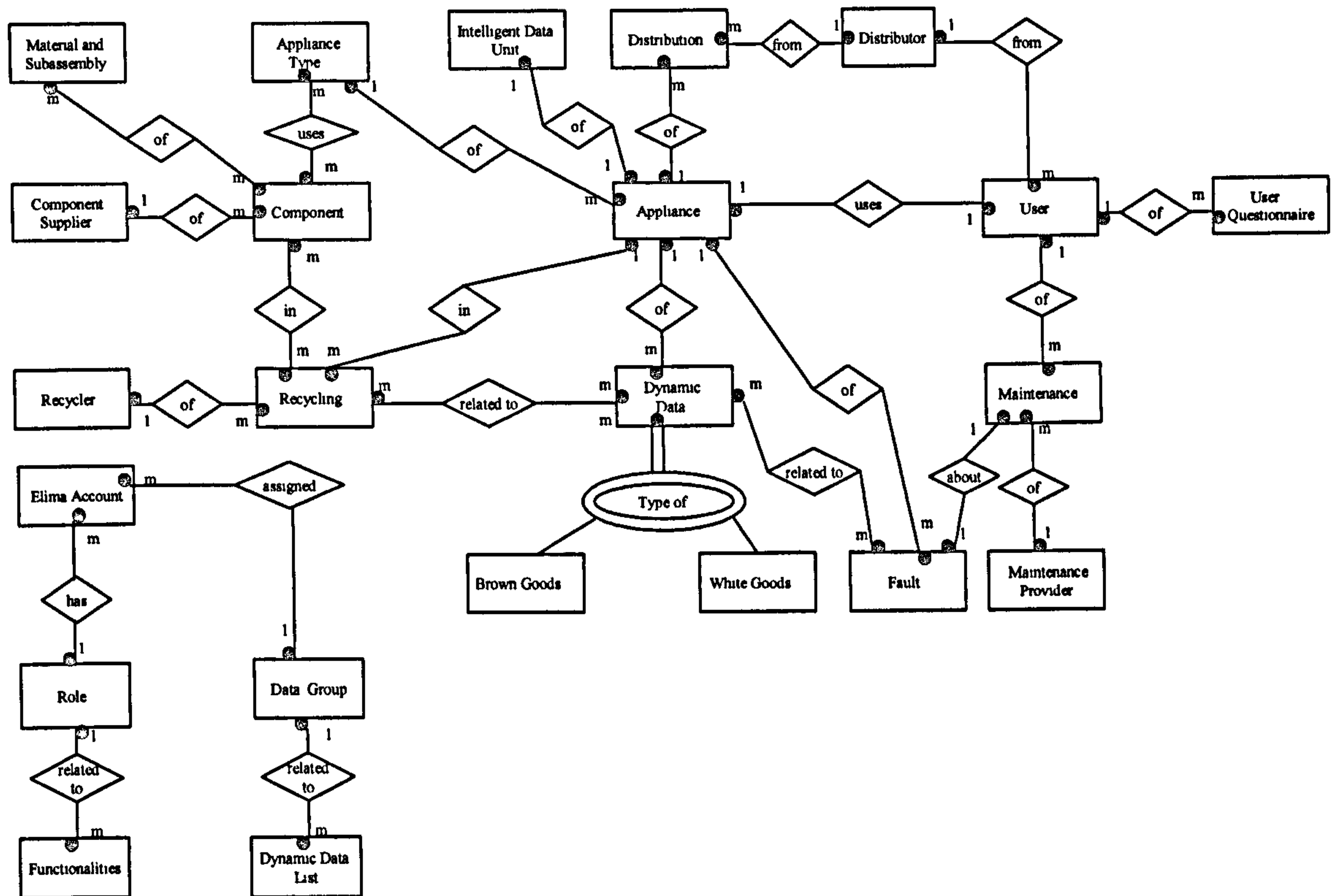


Figure 6.11 Life cycle database E-R diagram

### 6.6.3 Life Cycle Database Schema

The life cycle database E-R model is just a high level representation of conceptual model; it has to be converted into a schema so that the database can be implemented on a given DBMS (Rolland, 1998). In order to support the database implementation, a database schema is developed based on the E-R model diagram as follows:

Appliance (Appliance-ID, Appliance-Type-ID, Serial-Number, Build-date, production- Line...)

Appliance-Type (Appliance -Type-ID, Type-Name, Appliance-Code, Appliance-Category, Length...)

Component (Component-ID, Component-Name, Supplier-ID, Drawing-Number...)

Component-Appliance-Type (Component-ID, Appliance-Type-ID...)

Material-and-Subassembly (MS-ID, Name, Hazardous, Valuable, Drawing-number...)

Component-Material-Subassembly (Component-ID, MS-ID...)



Component-Supplier (Supplier-ID, Supplier-Name, Address, Contact Number...)  
 Recycler (Recycler-ID, Name, Address, Contact Number...)  
 Recycling (Recycling-ID, **Recycler-ID**, **Appliance-ID**, Description...)  
 Recycling-Component (Recycling-ID, Component-ID...)  
 Recycling-Dynamic-Data (Recycling-ID, Dynamic-Data-ID...)  
 Brown-Goods-Dynamic-Data (Dynamic-Data-ID, **Appliance-ID**, Collect Date, Collect time...)  
 White-Goods-Dynamic-Data (Dynamic-Data-ID, **Appliance-ID**, Collect Date, Collect time...)  
 Fault (Fault-ID, **Appliance-ID**, Fault Description...)  
 Fault-Dynamic-Data (Fault-ID, Dynamic-Data-ID...)  
 Maintenance (Maintenance-ID, **Fault-ID**, **User-ID**, **Maintenance-Provider-ID**, Description...)  
 Maintenance-Provider (Maintenance-Provider-ID, Name, Address, Contact Number...)  
 User (User-ID, **Appliance-ID**, **Distributor-ID**, Name, Purchase Date, Appliance Location...)  
 Questionnaire (Questionnaire-ID, **User-ID**...)  
 Distribution (Distribution-ID, **Distributor-ID**, **Appliance-ID**, Date, Time...)  
 Distributor (Distributor-ID, Name, Address, Contact Email...)  
 Elima-Account (Elima-Account-ID, **Data-Group-ID**, **Role-ID**, Email Address, Password...)  
 Data-Group (Data-Group-ID, Data Group Name...)  
 Role (Role-ID, Role Name...)  
 Functionalities (Functionality-ID, **Role-ID**, Functionality Name ...)  
 Dynamic-Data-List (Dynamic-Data-List-ID, **Data-Group-ID**, Dynamic Data Name...)

The life cycle database is designed to contain 26 relational tables. The primary key for each table is highlighted with underscore; while the foreign key is highlighted in bold font. The innovative features of the life cycle database include:

- Data from different life cycle stage are brought together.
- Stakeholders involved in the product life cycle are brought together.
- Product related data are integrated together.

## 6.7 Class Libraries and APIs

In order to support the creation of product-specific service enabler, the Framework provides a set of class libraries and APIs, which are developed based on the life cycle data structure schema and life cycle database schema. The object-oriented methods of software engineering are employed in the design and development. Major class libraries and APIs provided by the Framework include:

- Database operation



- Excel spreadsheet generator
- HTML table generator
- Chart generator
- Dynamic data definition
- File name operation
- File folder operation
- Security management
- Statistical analysis functions
- Dropdown menu builder

The design of class libraries and APIs is too vast to include in the thesis. However, the description of major class libraries and APIs can be found in Appendix A.

## 6.8 Components Identification and Categorisation

According to Short (1997), the software framework can be designed to provide a library of interconnecting components that could be linked together using a scripting language. The software framework aims to ensure the system developer can reuse the generic components and can concentrate on the product-specific algorithm or functional design. Therefore identifying components and categorising them into generic components and product-specific components are crucial in the development of the Framework. The generic components have implementations but the product-specific components only provide specifications. The service enabler developer can provide the code of business logic to the product-specific component based on the interfaces defined by the specification. Based on the IMS use case analysis and three-tier distributed architecture, components within the Framework are identified as follows, as shown in Figure 12.

In terms of component deployment, the component can be categorised into server-side components and client-side components. Service-side components usually provide functional capabilities while client-side components are concerned with a user-friendly interface.



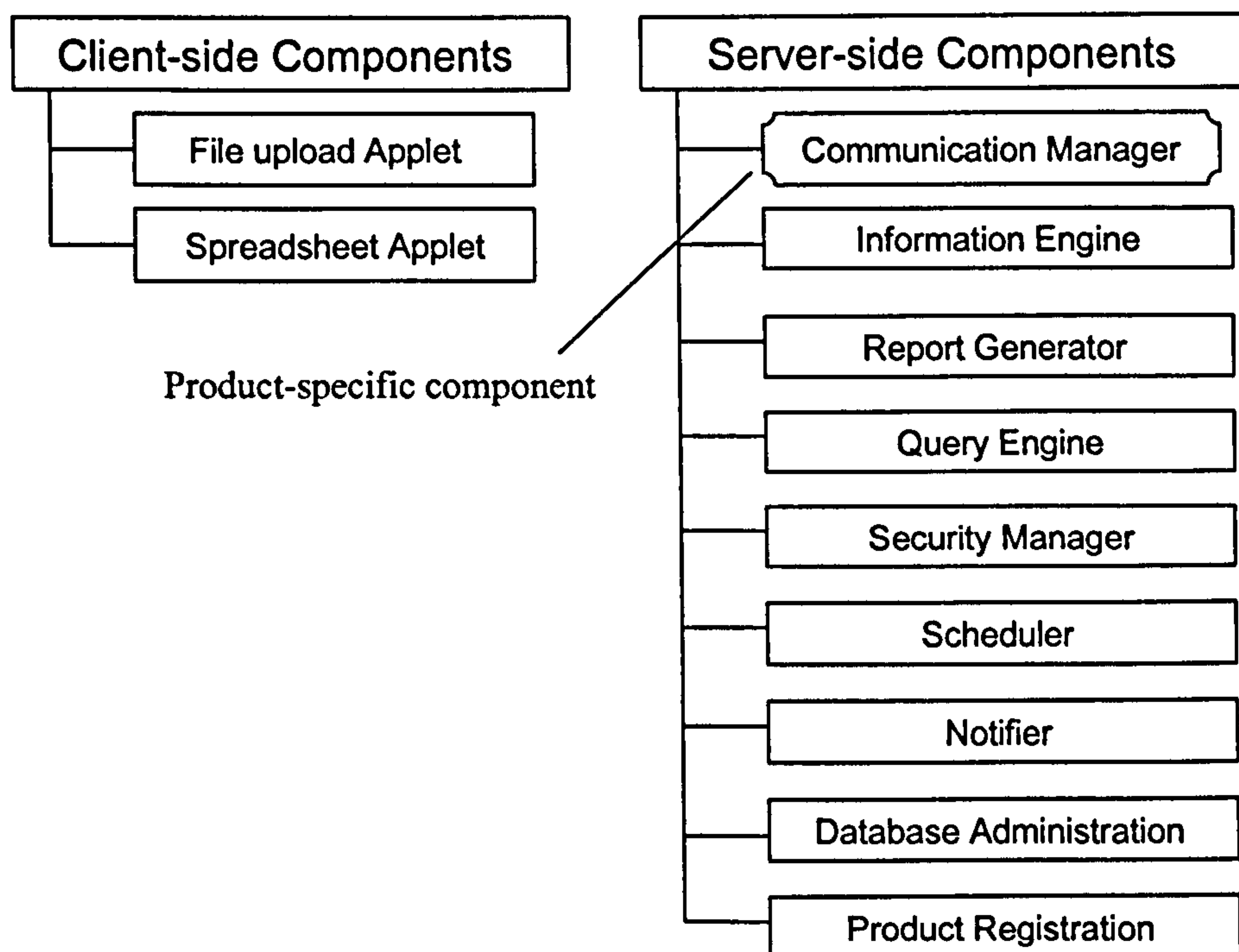


Figure 6.12 Components hierarchy

Among the identified components, generic components include:

- *Information Engine*
- *Report Generator*
- *Query Engine*
- *Security Manager*
- *Scheduler*
- *Notifier*
- *Database Administration*
- *Product Registration*
- *File Upload Applet*
- *Spreadsheet Applet*

The product-specific component is *communication manager*.



## 6.9 The Design of Communication Manager

As a component part of service enabler, *communication manager* is responsible for receiving life cycle data from various channels (e.g. raw data from IDU and import BOM data from manufacturer) and feeding them into the database. The *communication manager* component within the Framework should be designed to support both file-based data transfer and live data transfer. Furthermore, the modular design of *communication manager* should be considered to enable subsequent replacement and/or expansion with minimal rippling effect on the existing system. The design specification of *communication manager* is as follows.

### 6.9.1 Neutral Format Layer

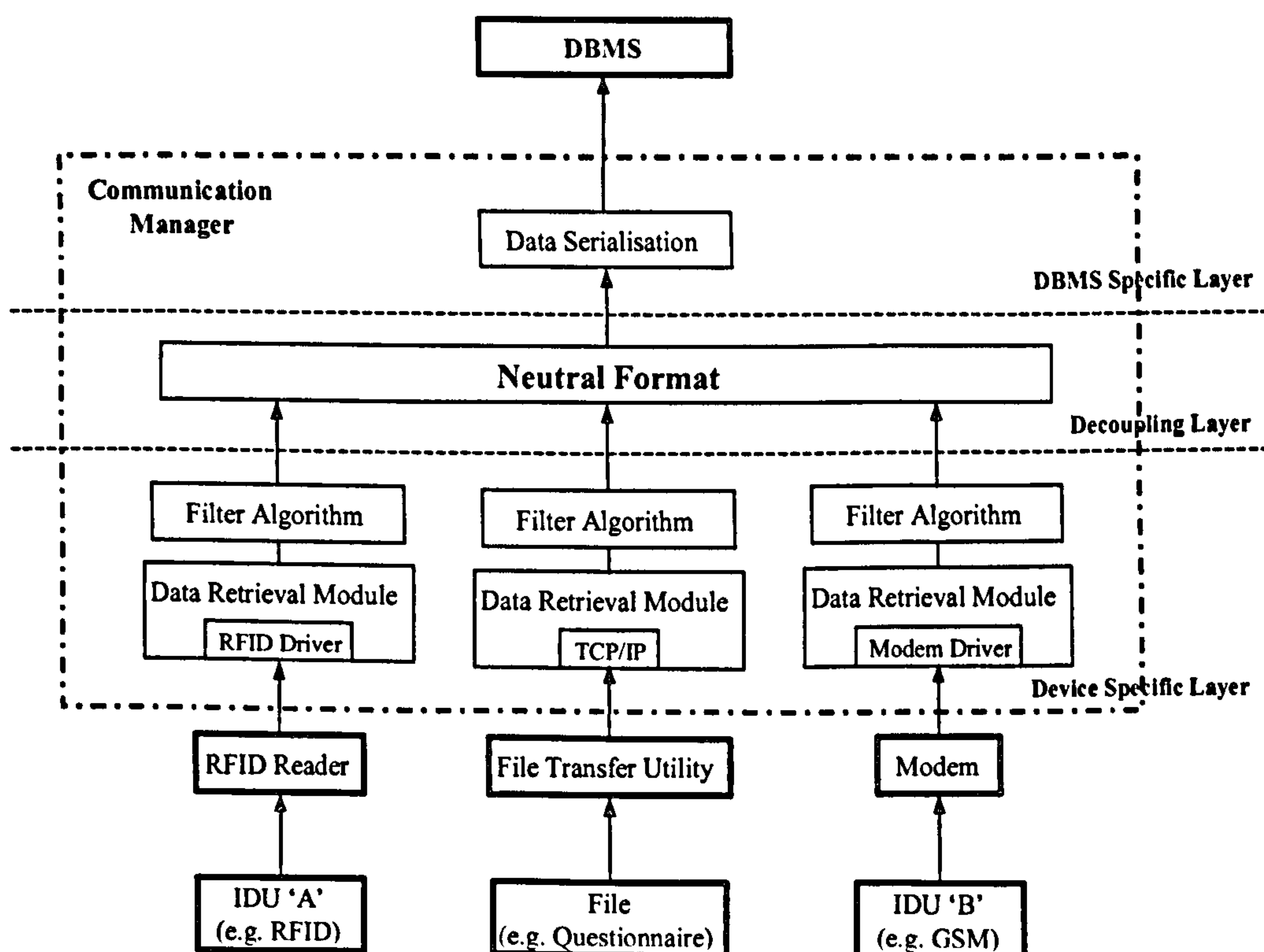


Figure 6.13 Communication manager neutral format layer

The acquired life data can be transferred to service enabler through various mechanisms such as *file upload*, *GSM modem*, etc. In addition, the data/information store can adopt various DBMS such as *MySQL*, *SQL Server*, etc. Once the service enabler is developed,



the adoption of other data transfer mechanism or migration to other database server will have impact on the whole system. In the worst scenario, this may cause the system to crash. In order to tackle these limitations as well as to allow service enabler to support different data transfer mechanisms and various DBMS, we employ a decoupling layer to separate the DBMS specific and device (i.e. IDU) specific implementation. The design concept of the communication manager is depicted in the Figure 6.13.

The concept behind the neutral format is the “plug-and-play”, which aims to minimise the ripple effect to the existing system when a specific component of the system is added or removed. The neutral format layer is designed to decouple the specific DBMS (i.e. MySQL, SQL Server) and specific IDU (i.e. RFID driver, GSM modem driver) to allow the subsequent data transfer mechanism or DBMS replacement or system extension without any impact to existing system. The XML technology together with the developed *dynamic data structure schema* is employed for the implementation of the neutral format, and a corresponding translation engine needs to be developed to parse the XML document.

### 6.9.2 File - based Data Transfer

File-based dynamic data transfer is an effective mechanism for transferring the life cycle data to the service enabler. In this transfer mode, data is collected and stored as XML or normal data file format (e.g. CSV) in a local computer. In order to transfer the data to the remote service enabler, a web-based file upload mechanism has to be provided for the file(s) to be uploaded from local computer to the remote service enabler. The mechanism should support the upload of multiple files by choosing files separately or collectively, or by just clicking an individual file folder containing data files to be uploaded. The operation sequence for uploading file(s) to the remote service enabler is depicted in Figure 6.14. The user chooses files to be uploaded using the file upload Applet provided. The received file is then verified against the corresponding data schema before data is extracted from the file. The extracted data is then stored in the database.



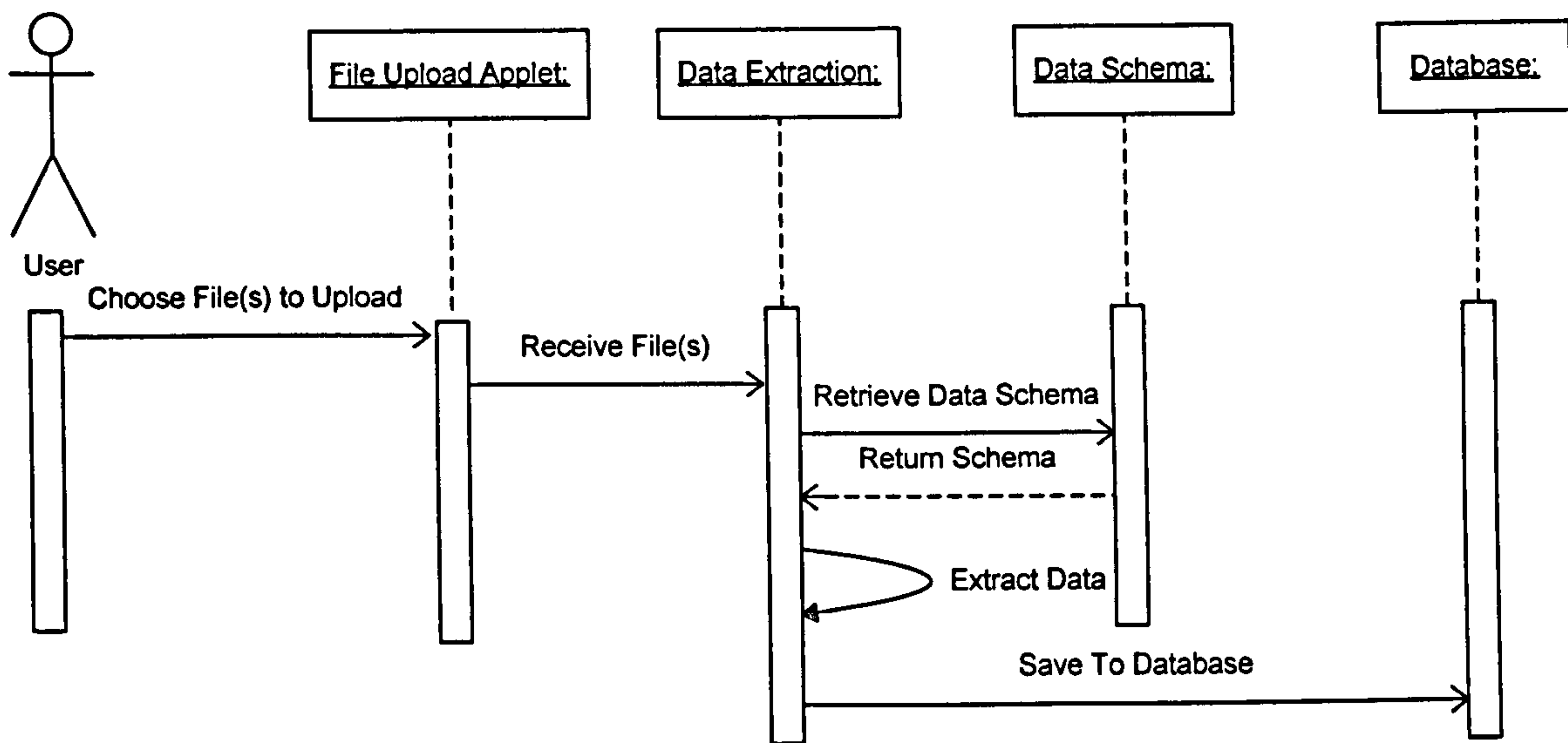


Figure 6.14 Sequence diagram of file upload operation

### 6.9.3 Live Data Transfer

The live data transfer features that life cycle data is constantly transferred into the service enabler in a form of data stream over the CSI. As the data is transferred lively, the size of the data and the interval between each data transmission are uncertain. They are random and dynamic. Another concern is: when the *Data Extraction* is busy in processing data stream, any other oncoming data stream may be blocked or lost. In order to tackle these problems, an Internet-based solution to support the live data transfer is proposed and shown in Figure 6.15.

This solution is based TCP/IP and adopts the concept of multi-thread processing. At the service enabler side, a communication port and socket needs to be created and bound to an IP address. Once completed, service enabler keeps listening to the port and waits for the connection coming from IDU. If a connection coming from IDU is detected, the service enabler will create *thread 1* which dedicates to read the data from stream and write them to the database; in the meanwhile service enabler itself carries on listening to the port and waits for the next connection. If another connection is detected, service enabler will create *thread 2* to handle the stream while itself still keep listening to port and wait for the next connection.



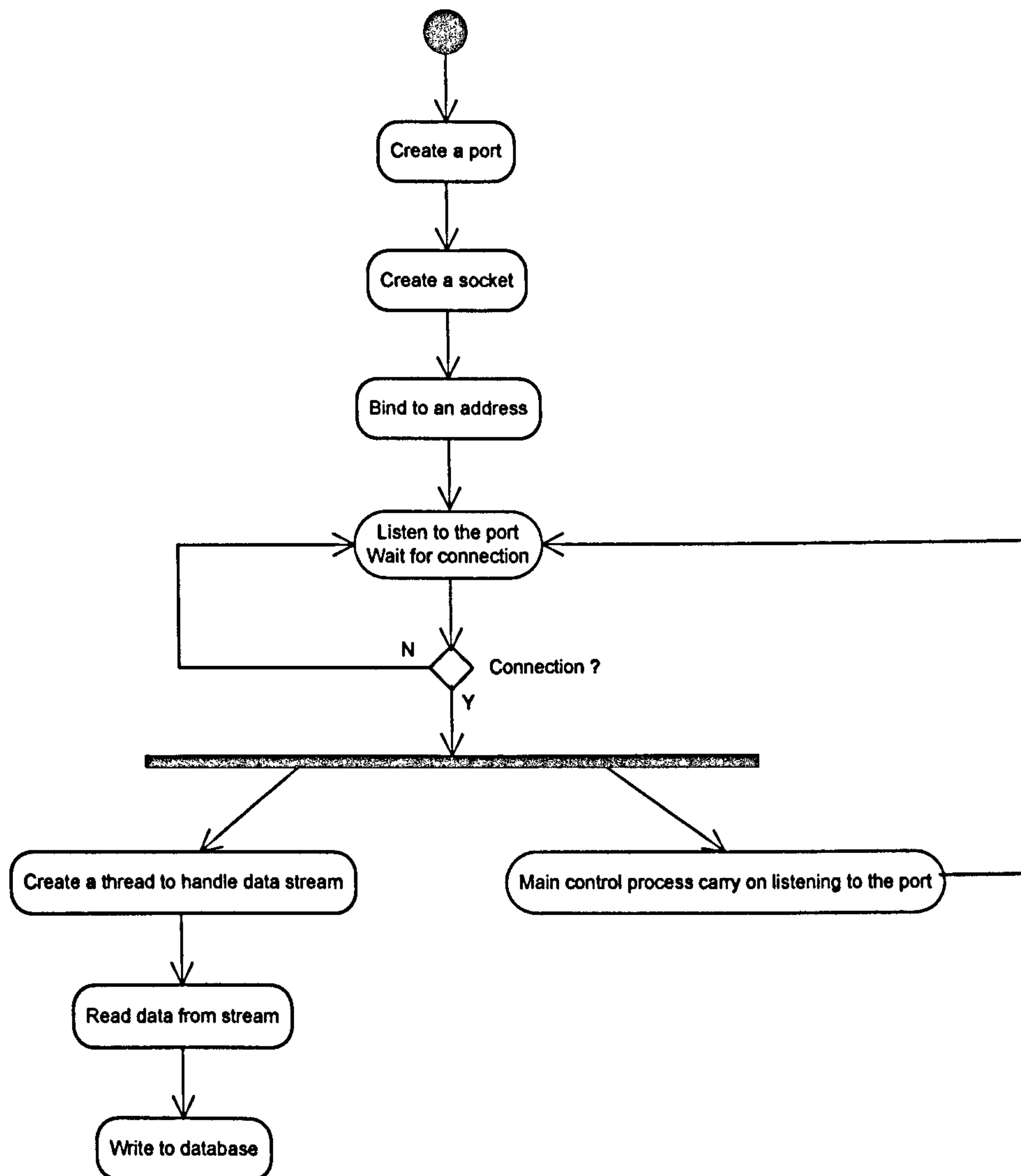


Figure 6.15 A solution for live data transfer

## 6.10 The Design of Security Manager

Security of data held by the service enabler is always paramount, especially for the sensitive life cycle data (e.g. commercial value, Data Protection Act). Considering this, only the normal *username-password* authentication mode for security management is not sufficient. Since all stakeholders would have equal privileges of access, certain users may be able to access data which is not relevant to them or nor intended for them. For example, a product end-user may access functionality like database administration. Therefore apart from common password authentication, the *security manager* component within the Framework should be designed to support the security control at a



functionality level and data level. In other words, the *security manager* should provide security control in three levels:

- log in access control
- data access control
- functionality access control

In order to provide full security to the service enabler, a security model which includes “*three-level access control mechanism*” and “*role-session paradigm*” is designed.

### 6.10.1 Three-level Security Control

A “*three-level security control mechanism*” is shown in Figure 6.16:

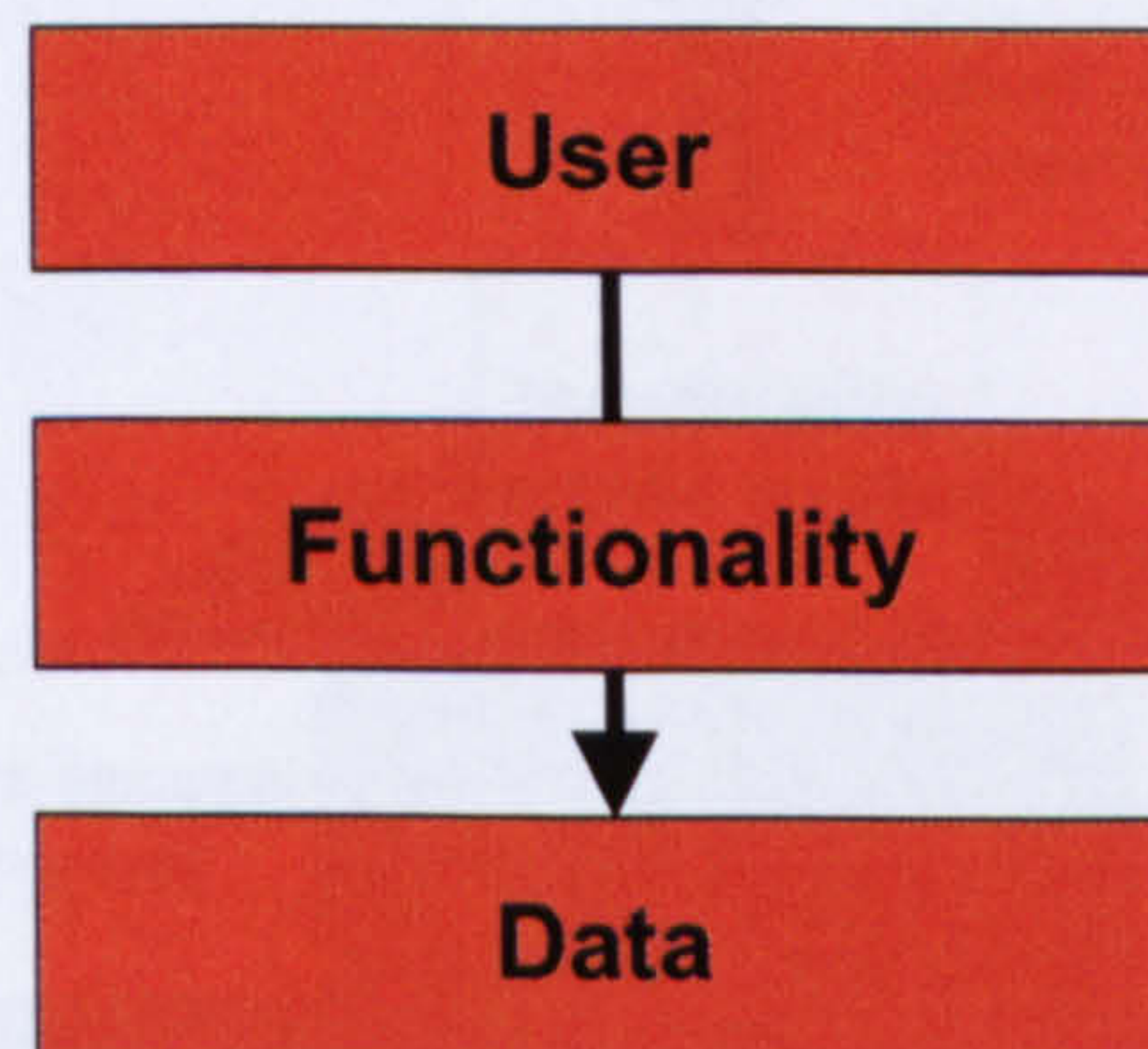


Figure 6.16 Three-level access control

- *Log in access control.* Only user who has valid password can log in.
- *Functionality access control.* Assign access privilege to functionality to avoid mishandling. For example, user management and DBMS management functionalities are only accessible to administrator.
- *Data access control.* Assign access privilege to data. For example, the role of recycler is only allowed to see part of the dynamic data that is related to recycling.



6.10.2 Role and Session Paradigm

The three-level security control mechanism is achieved by employing the “*role and session paradigm*”. This paradigm is described as follows: The authorised user will be granted a password and a role. The password is used for login access control, while the role is closely associated with a session, which determines the level of functionality access and data access. Once a user who plays a certain role logs into the system, a corresponding session is created and what functionalities and data can be accessed is specified.

When a user logs on to the system, the three-level access control mechanism is invoked, and a new “session” based on the “role” the user plays is created. The “*role and session*” paradigm for the security control is depicted in Figure 6.17.

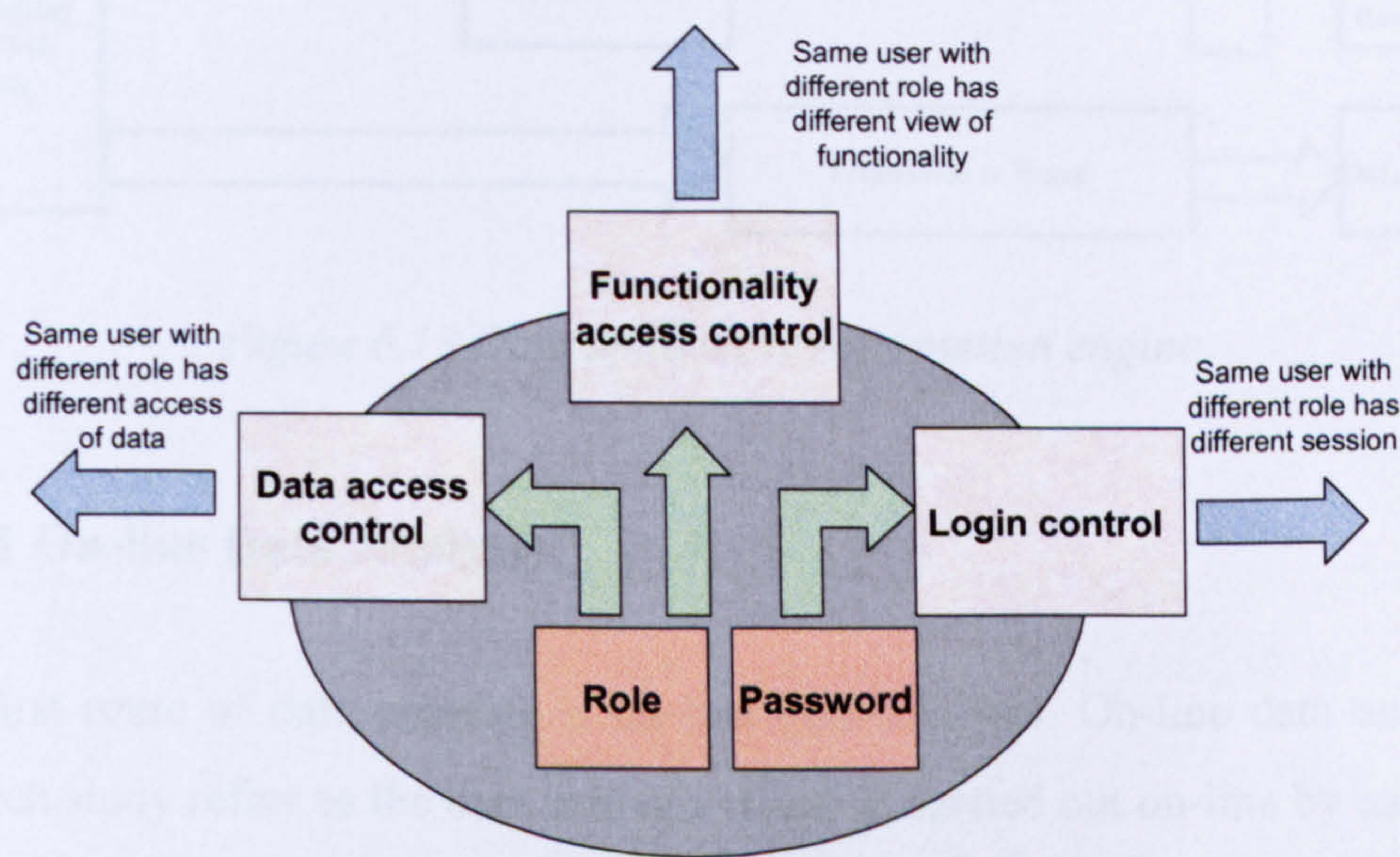


Figure 6.17 Service enabler security model

The *security manager* pre-defines a series of roles such as administrator, design & development, recycler, etc. Each role is associated with the security profile, which is a collection of functionality access privilege level and data access privilege level. When establishing a user login account, the password will be required to create and the role will be required to assign. One user can play different roles, which means the same user with different roles has a view of different functionalities and data.



### 6.11 The Design of Information Engine

Considering ELIMA field trial features and requirements, *information engine* component within the Framework is designed to provide two mechanisms for data quantitative analysis: (i) on-line data analysis (ii) transformation of data into Excel spreadsheet, as shown in Figure 6.18.

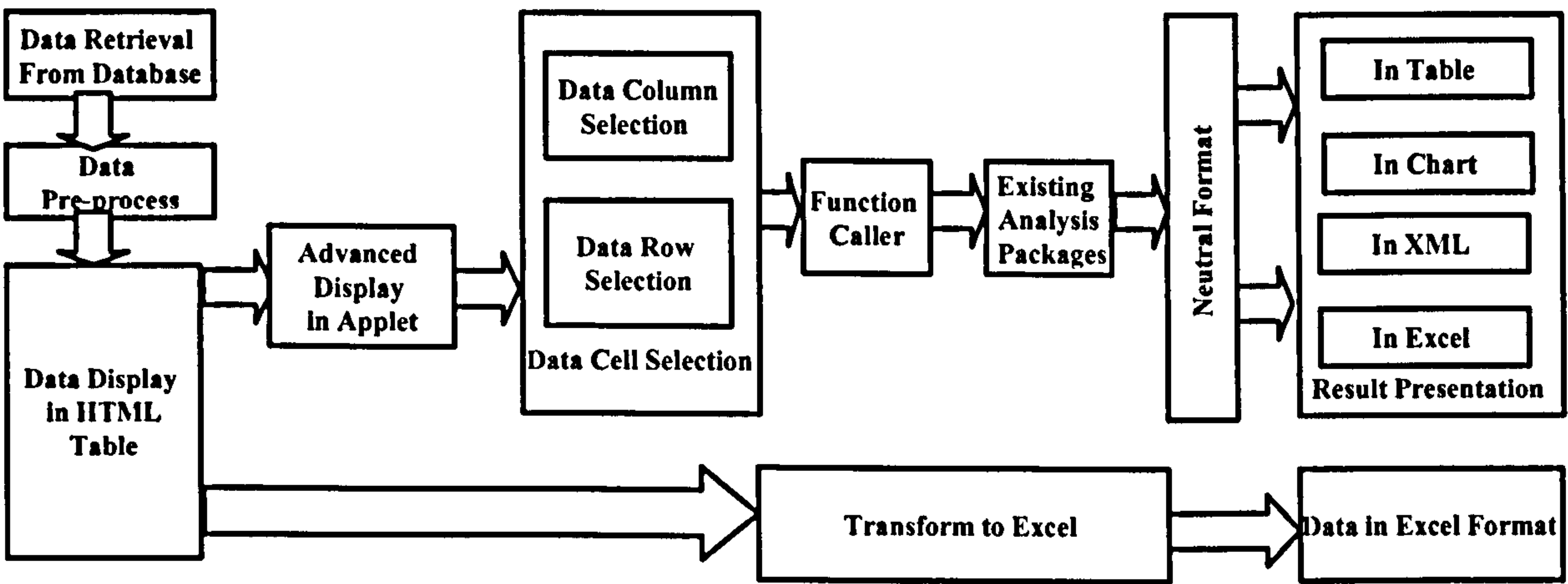


Figure 6.18 Data analysis in information engine

#### 6.11.1 On-line Data Analysis

The first route of data analysis is on-line data analysis. On-line data analysis in this research study refers to the data analysis which is carried out on-line by using statistical functions and analysis tools provided the service enabler. This means that the *information engine* should provide the capabilities to allow users to selectively choose the dataset for analysis, and itself should have a built-in library of statistical functions and analysis tools. A use case diagram of on-line data analysis is created to capture the requirements, as shown in Figure 6.19.



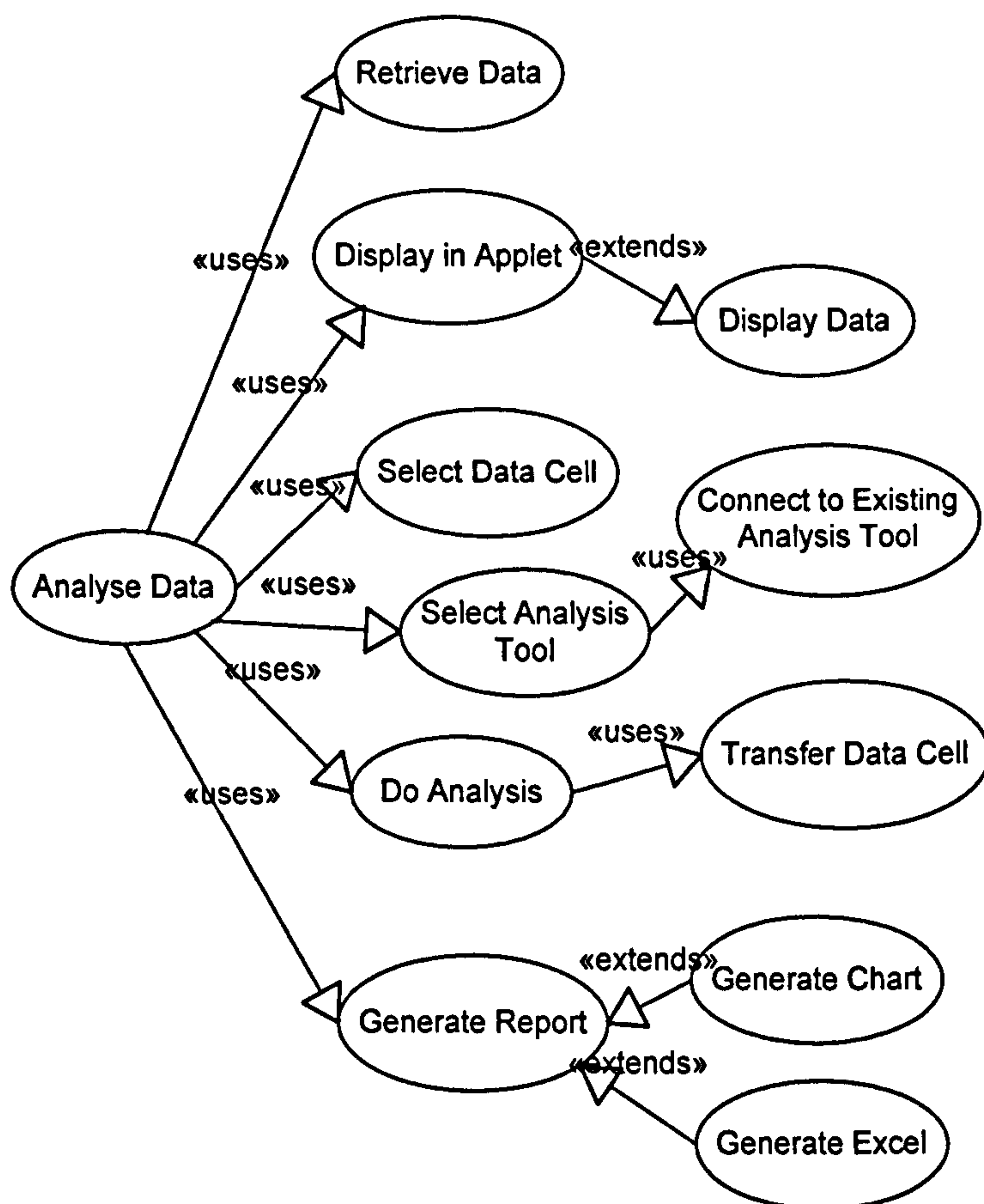


Figure 6.19 On-line data analysis

One main challenge of the on-line data analysis is: once data is displayed as a table in an HTML web page, how can we flexibly choose the dataset based on rows or columns? The reason of this challenge is that HTML tabular data does not support data random selection and data sorting, etc. In order to overcome this limitation, a Java Applet technology is employed to display the data in an Excel look-and-feel spreadsheet, where manipulations such as data random selection and data sorting, etc can be carried out by using the mouse. It also supports the whole data column shift to the right or left for easier data comparison and selection. The selected data is then passed to *Function Caller* which processes the data with analytical routines. The analysis results are then transformed into a neutral format, ready to be presented in other appropriate format for users or applications (See Figure 6.18).

Another main concern of on-line data analysis is how the *information engine* can provide a library of statistical functions and analysis tools. It is an option to develop each analysis tool and statistical function (e.g. correlation) from scratch. But this is an



awkward solution. As many statistical functions and analysis tools are standard routines (e.g. correlation, histogram) which can be found in some existing analysis and simulation software packages (e.g. Excel, SPSS, MatLab), a smart solution of developing this library may be to incorporate the existing statistical functions and analysis tools into the *information engine*.

Implementation details of *information engine* using Java Applet to display life cycle data in Excel look-and-feel spreadsheet, and provision of the library of analysis tools and statistical functions can be found in Chapter 7.

### 6.11.2 Transformation of Data into Excel Spreadsheet

The second route for data analysis is *transformation of data into Excel spreadsheet*, i.e. to pre-process data such as sorting, grouping, and then transform the pre-processed data into Excel spreadsheet format where further analysis can be carried out by using built-in analysis tools and statistical functions within Excel. The generated Excel spreadsheet can be saved on a local PC and data analysis can be conducted off-line. This is an indirect data analysis method but quite useful. The sequence diagram of transformation of data into Excel format is shown in Figure 6.20.

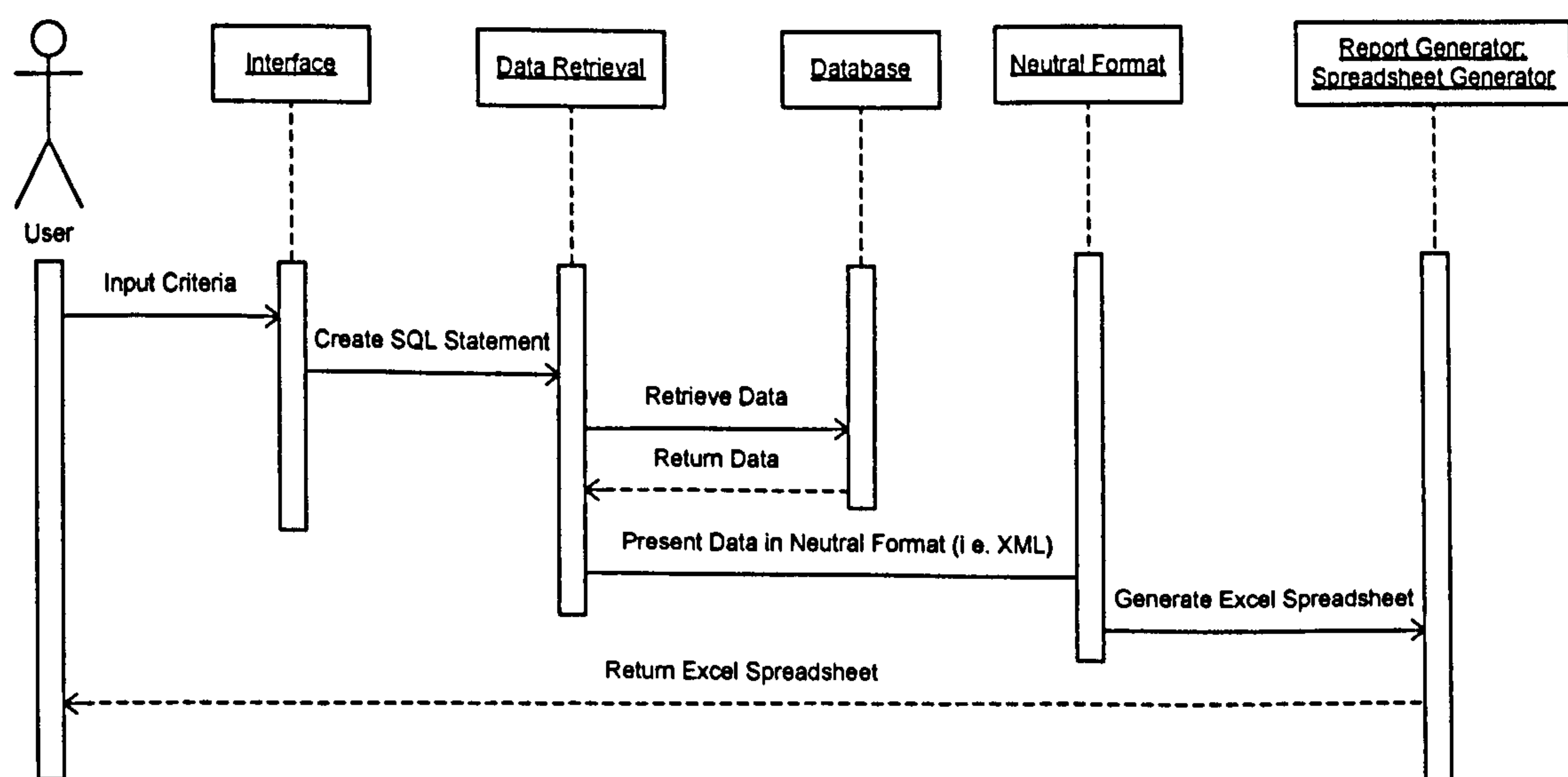


Figure 6.20 Transformations into Excel spreadsheet



As shown in the diagram, user creates the data retrieval criteria first. For example, to select all product units which are sold in Germany. After the retrieval criteria is created, the *Data Retrieval* is responsible for generating SQL statement, and sends the query message to *Database*. *Database* is responsible for retrieving the data and returning to *Data Retrieval*. Before generating the Excel spreadsheet, usually the retrieved data is transformed into an XML neutral format so that other applications or users can use this retrieval result. Finally the Excel spreadsheet of the retrieval data is generated by *Report Generator*.

Both data analysis methods complement each other, *on-line data analysis* provides instant processing and display of results. *Transformation to Excel format* allows users to use some comprehensive statistical software packages to analyse the data provided.

## 6.12 The Design of Query Engine

A query or search utility is designed to help user to find the required data, information or service content. The *query engine* component within the Framework is designed to support the query by one of the following or combination:

- Product model
- Serial number
- Dynamic data field
- Static data
- Date

Keyword searching is a commonly used way to traverse the database to find the keyword exactly matched or fuzzily matched data or information. Database Structured Query Language (SQL) is employed. SQL was originally devised in the 1970s as a language for relational database and since become the *de facto* industry standard language for relational systems (Rolland, 1998; Connolly et al, 2002).



### 6.13 The Design of Scheduler

*Scheduler* component within the Framework is designed to carry out predefined tasks (e.g. data storage, report generation) at set intervals (e.g. daily) or respond to predefined events (e.g. after receiving data). The scheduling process is shown in Figure 6.20. It proactively carries out the scheduled tasks; triggers notification to users when alert is necessary and logs results into database.

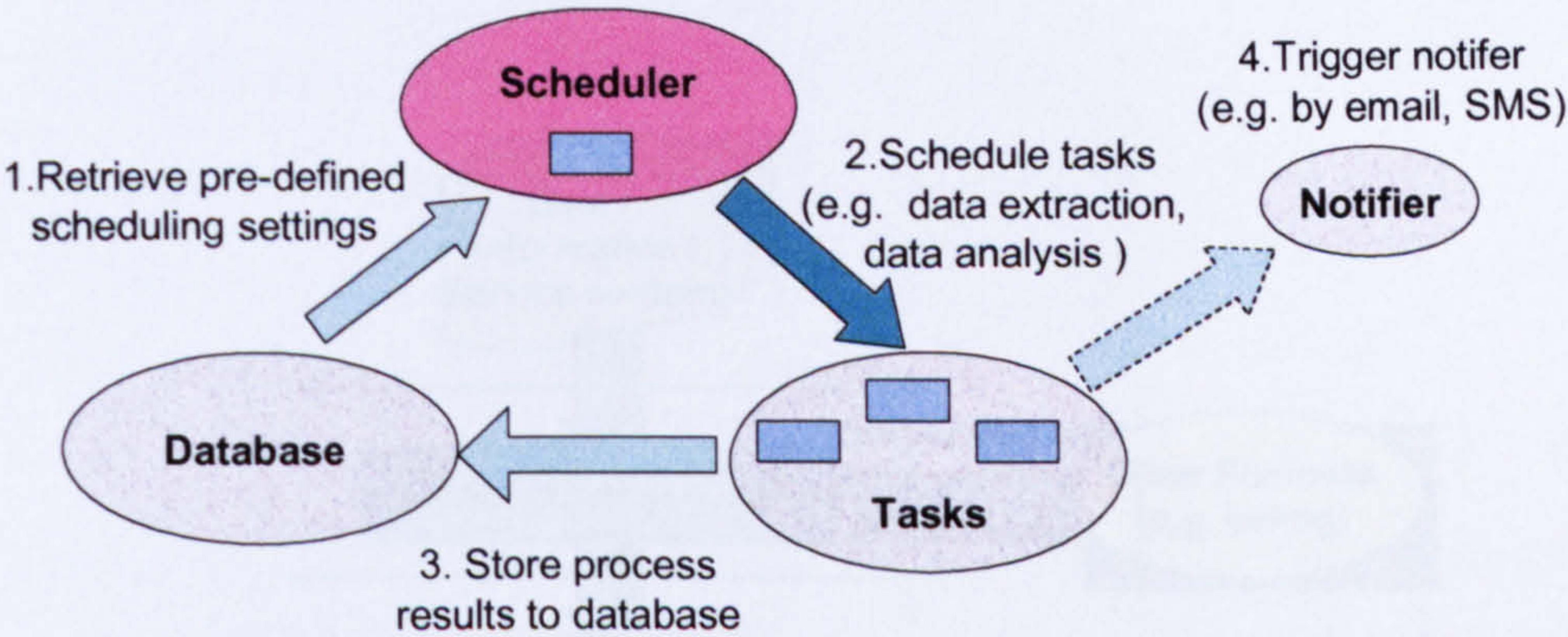


Figure 6.21 Task scheduling

### 6.14 The Design of Notifier

The service enabler should support user notification when attention is needed on the product, for example, when certain parameter exceed limits, or under predefined conditions as suggested by the manufacturer or service maintenance team. This is a cardinal function as this communication could be of great significance to the end user, who may take suitable action. The *notifier* component within the Framework is primarily designed to support the email notification.

### 6.15 The Design of Report Generator

The *report generator* component within the Framework is designed to present the requested data, information or service content in a structured presentation, for example, table, chart and Excel, etc, as shown in Figure 6.21. In order to support the share of data,



information or service content with others, the data, information or service content is transformed into XML document in the first instance so that they can directly communicate with third party's applications (e.g. bill calculating application uses energy consumption data to generate bills). As XML document has become a *de facto* data presentation standard, many business applications have the built-in utilities to interpret the XML document. After data, information or service content is presented in XML document, *report generator* transforms this XML document into the required format such as text, intuitive tables and charts, Excel spreadsheet, etc.

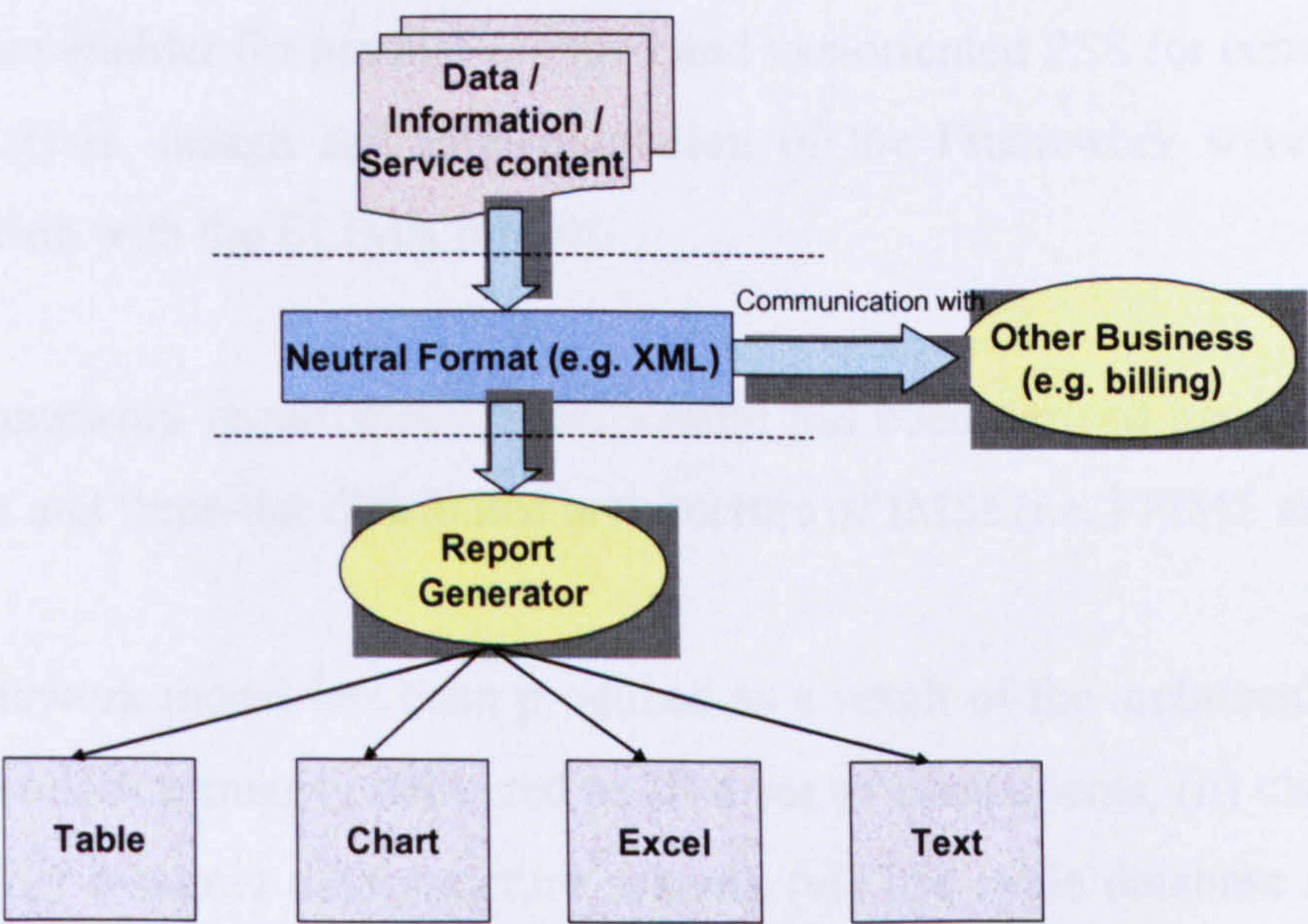


Figure 6.22 Report generator

### 6.16 The Design of Product Registration

User information (e.g. house hold number, age), product installation data (e.g. kitchen, garage) are also part of life cycle data. However, the transfer of static data and dynamic data does not cover this aspect. Such data can be regarded as a separate one which does not relate directly to the product.

The *product registration* component within the Framework is designed to enable user (e.g. product administrator) to register their products once in use. From the perspective of the registered users, they normally benefit from technical support (e.g. via email or



support request form), free product updates (if any) and product upgrade discounts (if available). Users also have the option to join a mailing list. From manufacturer's view, the *product registration* should be designed to help collect additional user information (e.g. country, number in the house hold, universal identity) and installation environment of the product (e.g. kitchen, garage).

## 6.17 Summary

- 1) The Framework has been formulated as a software toolkit to facilitate the realisation of service enabler for product-oriented and use-oriented PSS for consumer products. The analysis, design and implementation of the Framework were carried out in association with the ELIMA project.
- 2) The Framework requirements specification has been derived based on the use case analysis and three-tier distributed architecture of IMSs (i.e. FFIMS and PSIMS).
- 3) A Framework model has been produced as a result of the architectural design. The Framework is primarily delivered as (i) a set of components, (ii) class libraries and APIs, (iii) dynamic data structure schema (vi) life cycle database schema and (v) runtime infrastructure.
- 4) The life cycle data design guideline has been produced to support the general life cycle data design for consumer products. The major contribution of this guideline is the development of dynamic data structure schema for consumer products, which can be used as a reference to help define the product-specific dynamic data structure.
- 5) The life cycle database schema for consumer products has been developed by employing database E-R modelling technique to support the database implementation.
- 6) The detailed design of each component within the Framework, namely, *communication manager*, *information engine*, *security manager*, *query engine*,



*scheduler, notifier, report generator* and *product registration* have been described respectively. The implementation details of them will be described in Chapter 7.



# Chapter 7 Implementation of the Framework and Supporting Tools

## 7.1 Introduction

This chapter describes the implementation of the Framework and associated supporting tools to facilitate the creation of service enabler for consumer products. The strategy of this implementation is to use the off-the-shelf components wherever possible

## 7.2 Development Tools and Implementation Strategy

### 7.2.1 Development Tools

The main development tools for the Framework are *Macromedia Dreamweaver MX* and *TextPad*.

- *Macromedia Dreamweaver MX* (<http://www.macromedia.com>) is a professional HTML editor for designing, coding, and developing websites, web pages, and web applications. Besides support building dynamic database-backed web applications using PHP, it also supports other popular server languages such as *ASP*, *ASP.NET*, *ColdFusion Markup Language (CFML)*, *JSP*. The *Dreamweaver* provides both visual editing environment and hand-coding environment. The visual editing features can let designers quickly create pages without writing a line of code. The *Dreamweaver* includes many coding-related tools and features to support the hand coding.
- *TextPad* (<http://www.textpad.com>) is a lightweight Integrated Development Environment (IDE) employed for Java component development. In comparison with other Java IDE such as *Jbuilder* and *WebSphere*, etc, it is lightweight yet free. Furthermore, it also has the following features for Java application/component development:



- Integrates Java compilation and execution
- Supports Java application and component (e.g. Applet, JavaBean) development
- Supports code track and debug

### 7.2.2 Implementation Strategy

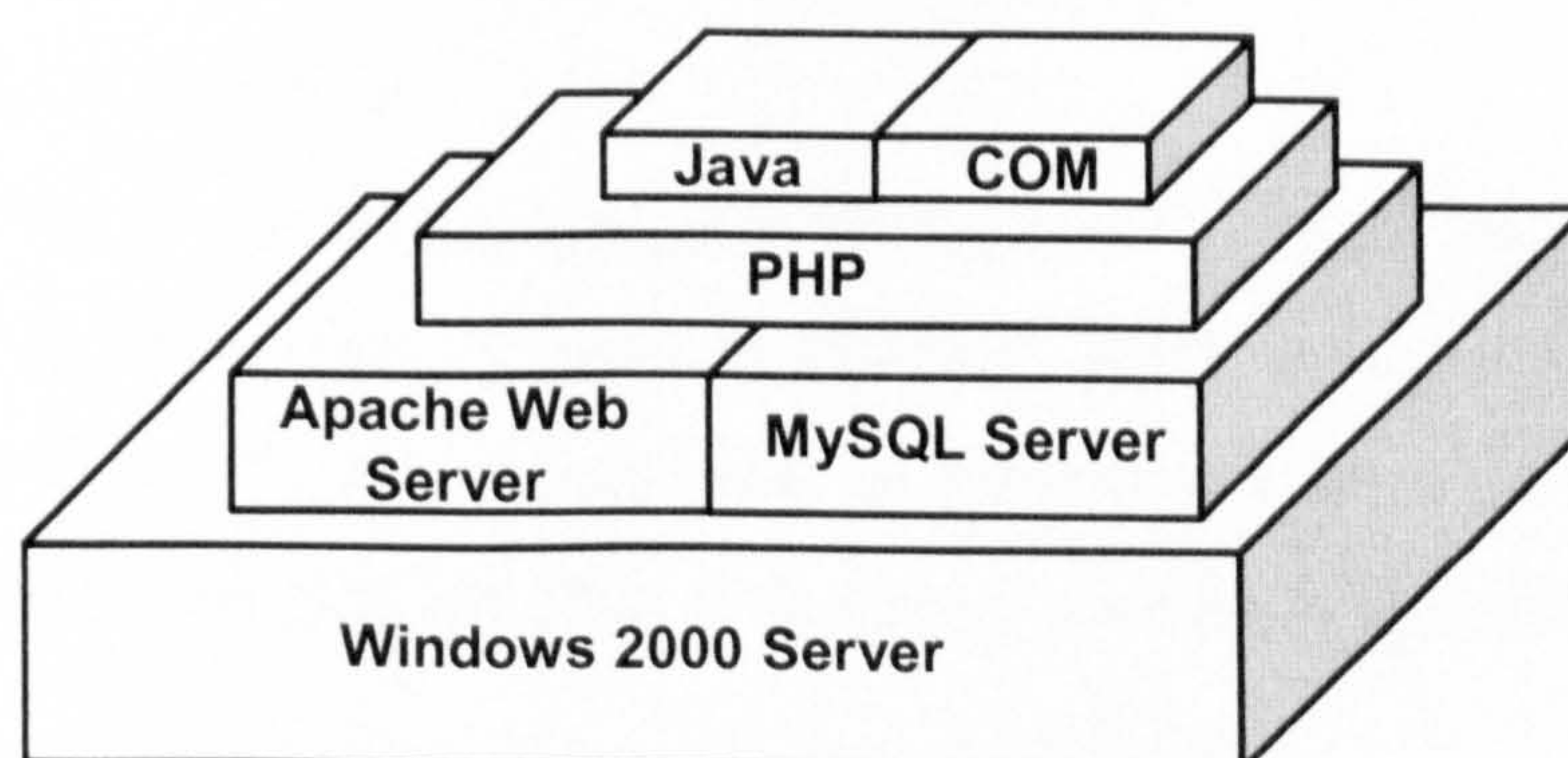
Developing each software component from scratch would be a rather tedious job, and is not the intention of this research study. Thus the strategy of implementing the Framework in this research study is to reuse the off-the-shelf software components wherever possible. The benefits of the reusing third partys' powerful components are stated as follows:

- It helps researchers concentrate on the research study instead of developing software.
- Many off-the-shelf software components are more powerful and professional than ones developed by researchers.
- Cost and time efficient. Many off-the-shelf software components are free and can be downloaded from the Internet.
- Component reuse is the trend of software development.

## 7.3 The Framework Implementation

### 7.3.1 Runtime Infrastructure

The runtime infrastructure within the Framework is built by employing the following products and technologies, as illustrated in Figure 7.1:



*Figure 7.1 Runtime infrastructures within the Framework*



- *Microsoft Windows 2000 Server* - The Local Area Network (LAN) of the IMAS laboratory at DMU largely adopts Microsoft Windows products
- *Apache Web Server* - PHP is chosen as a server side scripting language as it is open source, free and lightweight
- *MySQL Database Server* – Open source and free relational DBMS
- *PHP Zend Engine* – To serve the PHP
- *Java Runtime Environment (JRE)* – To Serve Java which is a powerful and free
- *Component Object Model (COM)* – Used to create the components which wrap the Excel worksheet functions of Microsoft.

The main feature of this runtime infrastructure is its cost efficiency. Except the *Microsoft Windows 2000 Server* and *COM*, other products and technologies are free and open source. This will significantly reduce the cost when realising a service enabler solution. The brief introduction of the products and technologies is as follows:

- Microsoft Windows 2000 Server

*Microsoft Windows 2000 Server* is a server operating system designed for mission critical applications (Windows Comparison, 2004). It has proven itself as a stable and reliable platform for Internet services. Besides the *Microsoft Windows 2000 Server* operating system, *Linux* has become a serious contender as a business operating system. Although *Linux* is predicted to have a bright future because it is free, the application are getting more mainstream and the GUI is pretty much like Windows, etc, *Microsoft Windows 2000 Server* are still very popular in many Small Manufacture Enterprises(SME).

- Apache Web Server

*Apache Web Server* (<http://www.apache.org>) has been the most popular web server on the Internet since April of 1996. In February 2005, *Netcraft Web Server Survey* found that more than 68% of the web sites on the Internet are using Apache, thus making it more widely used than all other web servers combined (Netcraft Web Server Survey, 2005).



The combination of *Linux*, *Apache Web Server*, *MySQL* and *PHP* is probably the most common production environment for Web applications (TopXML, 2004). They are now widely used in many SMEs, business and academic research due to the free, powerful functionalities, open source code and ease of use.

- PHP Zend Engine

The *Zend Engine* is the centerpiece of PHP, and is the component that parses and executes PHP files. PHP Zend engine is a core in PHP as it is responsible for (Zend Engine, 2005):

- High performance parsing (including syntax checking), in-memory compilation and execution of PHP scripts
- Implementation of all of the standard data structures of PHP
- Interfacing with extension modules for connectivity to external resources and protocols, such as SQL, HTTP and FTP
- Overloading the Object Oriented syntax, for integration with Java and .NET
- Providing all of the standard services, including memory management, resource management and more for the whole of PHP

- MySQL Database Server

The *MySQL* (<http://www.mysql.com>) database server is the world's most popular open source database. Over six million installations use *MySQL* to power high-volume Web sites and other critical business systems (Mysql, 2005). Reasons of choosing *MySQL* database server as DBMS are summarised as follows:

- Multithreading capabilities that enable the database to perform multiple tasks concurrently, allowing the server to process client request efficiently.
- Built-in support within PHP
- Open source
- The capability to handle large amount of data

- Java Runtime Environment

The *Java 2 Runtime Environment* (JRE) is the minimum standard Java computing environment for running applications written in Java programming language. It contains



the Java virtual machine, Java core classes, and supporting files. But it does not contain any of the development tools or classes that pertain only to a development environment (JAVA JRE, 2005).

- **COM Technology**

Microsoft *Component Object Model* (COM) technology in the Microsoft Windows-family of operating systems enables software components to communicate. COM is used by developers to create re-usable software components, link components together to build applications, and take advantage of Windows services (Microsoft COM, 2005).

### **7.3.2 Information Engine**

Implementation of *information engine* includes developing an Excel spreadsheet Applet and providing a library of analysis tools and statistical functions.

#### **7.3.2.1 Excel Spreadsheet Applet**

In order to overcome the limitation that HTML tabular data does not support data random selection and data sorting, etc, an Excel spreadsheet based data display is implemented by employing Java Applet technology. Figure 7.2 is a screenshot of Excel spreadsheet-based data display where data can be selectively chosen by dragging mouse, and statistical functions and analysis tools (e.g. histogram, correlation) can be applied to the selectively chosen data cell. Advantages of using spreadsheet-based data display are explained as follows:

- Support the use of various of statistical functions and analysis tools provided by service enabler
- Selectively choose data cell by dragging mouse
- Shift data column freely for comparison
- Sort data by data column



Advanced Data Display Customised Display

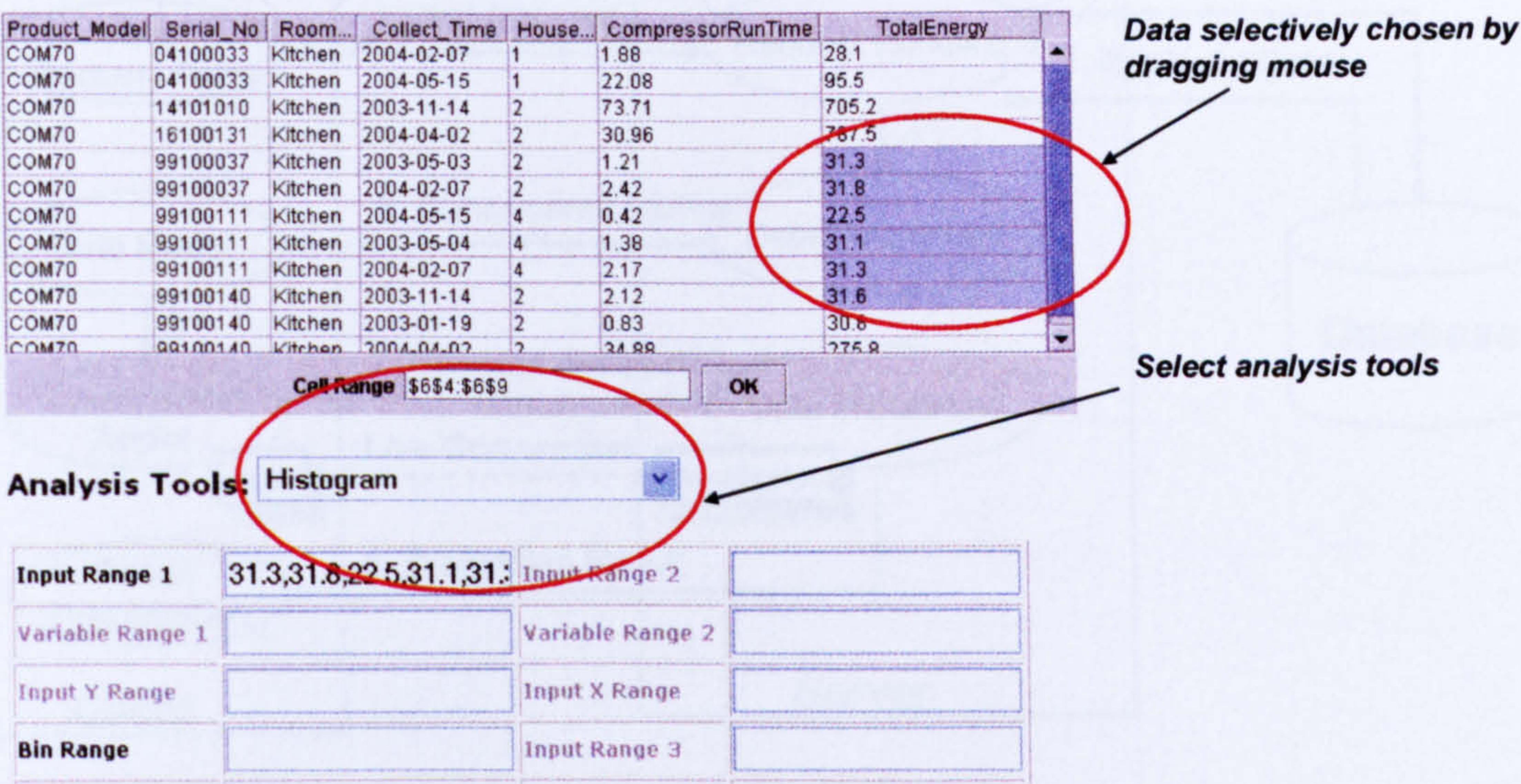


Figure 7.2 A screenshot of spreadsheet-based data display Applet

The operation process of the using spreadsheet Applet for on-line data analysis is depicted as follows (Figure 7.3):

1. User's request for viewing data is passed to *request handling*.
2. *Request handling* handles user request, and sends user query message to *database*
3. The retrieved data is returned to *request handling*.
4. *Data conversion* converts arrays of data into the concatenated string for stable transmission.
5. The concatenated string is passed to *data model*, where the concatenated string is spilt and stored as two-dimensional array.
6. The data stored in two-dimensional array is transferred from *data model* object into the *Java table of Applet*.
7. The selectively chosen data cell at *Spreadsheet Applet* is passed into *data processing*.
8. The processing result which is presented in XML document (i.e. neutral format) is sent to *result presentation* where the XML document is transformed into the required format such as HTML table, chart, etc.



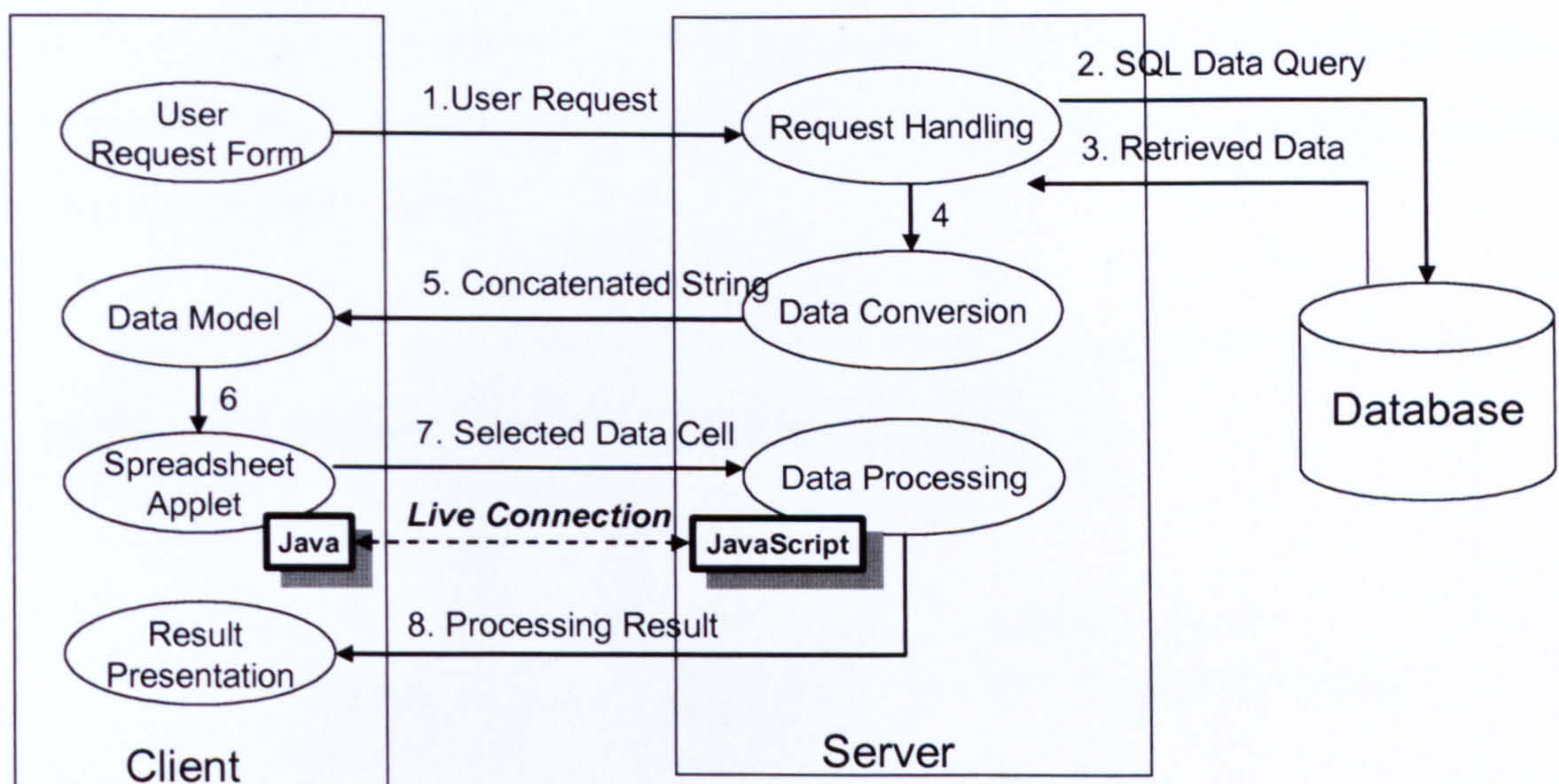


Figure 7.3 The process of on-line data analysis

### 7.3.2.2 Java-JavaScript Live Connection

The major challenge of this implementation is how the selectively chosen data cell at *spreadsheet Applet* is passed into *data processing*. This involves a general issue of message passing between the client-side Java Applet and server-side script (e.g. PHP, ASP). Normally HTML supports message passing from server-side script to client-side Java Applet by using an HTML tag *PARAM*. But this is the one-way message passing, i.e. message can only be passed from server-side script to client-side Applet, it does not support message passing from Applet to server-side script. There is no HTML tag which can achieve this. In order to tackle the problem of passing the selected dataset from client-side Applet to service-side *data processing*, an off-the-shelf component from the Netscape Company, namely, *LiveConnect* is adopted.

*LiveConnect* is a mechanism that allows server-side JavaScript to communicate with the Java virtual machine running at the client-side Web browser in a web page. Netscape's name for this new JavaScript-to-Java and Java-to-JavaScript communication facility is called "LiveConnect" (Flannagan, 1997). It is an underlying communication framework that opens up all sorts of possibilities for communication among JavaScript programs, Java applets, and Java-enabled plug-ins. *LiveConnect* can be thought of as the glue that



ties these things together (Figure 7.4). Through using *LiveConnect*, the selected dataset in *Spreadsheet Applet* can be easily transferred to JavaScript in a webpage, and then sent to *data processing* object.

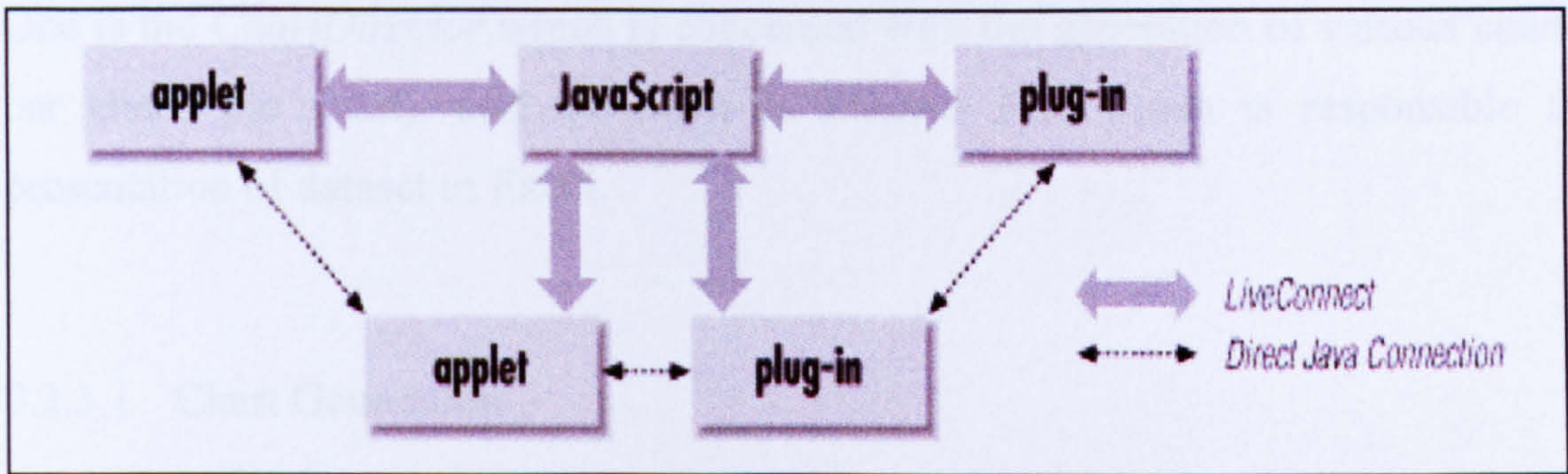


Figure 7.4 Live connect (Flannagan, 1997)

7.3.2.3 Provision of on-line analysis toolkit

The on-line analysis toolkit is implemented by reusing statistical analysis functions and analysis tools provided by *Microsoft Excel*. The main reason we chose to incorporate statistical analysis functions and analysis tools from *Microsoft Excel* into the *information engine* is that *Microsoft Excel* is a popular and powerful data analysis package. The reuse is achieved by developing a wrapper to connect to the Excel worksheet functions<sup>4</sup>, as shown in Figure 7.5. *Microsoft COM* technology is adopted in the development of *Worksheet Functions Wrapper*.

The class of statistical analysis functions implemented by wrapping the Excel worksheet functions can be found in Appendix A. (See A.2.4 *ElimaAnalysisFunctions*)

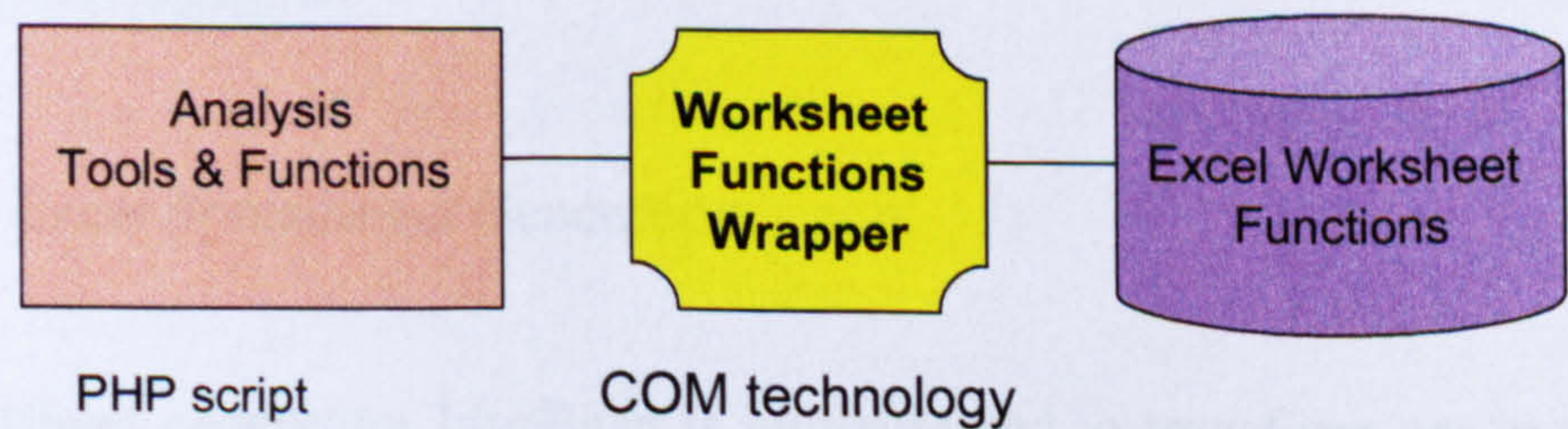


Figure 7.5 Reuse of Excel worksheet functions

<sup>4</sup> Microsoft Excel contains many predefined, or built-in, formulas known as worksheet functions (e.g. SUM, AVERAGE).



### 7.3.3 Report Generator

The *report generator* includes table generator, chart generator and Excel spreadsheet generator. Two off-the-shelf components are used to implement the *report generator*. One is the *ChartDirector* which is concerned with the generation of various charts (e.g. bar chart, pie chart), and the other is *Jakarta POI* which is responsible for the presentation of dataset in Excel.

#### 7.3.3.1 Chart Generation

*ChartDirector* (ChartDirector, 2005) is a charting and graphics component library for Web servers and for embedding into the Web applications. Reasons of choosing *ChartDirector* component is summarised as follows:

- Fast and efficient: Multi-threaded architecture specially designed for the demanding requirements of server side usage.
- Flexible: Object oriented API allows developer to control and customise chart details.
- Comprehensive chart styles: Pie, bar, line, step line, trend line, curve-fitting, and 3D effect, etc.
- Easy to use: Producing charts requires just a few lines of code. Comprehensive documentation and numerous sample programs are provided.

The class of chart generator developed by incorporating the *ChartDirector*, and the class of table generator can be found in Appendix A. (See A.2.5 *ChartGenerator* and A.2.2 *TableGenerator*).

#### 7.3.3.2 Excel Spreadsheet Generation

A spreadsheet generation JavaBean is implemented to transform product life cycle data into Excel spreadsheet by employing *Jakarta POI*. A *Jakarta POI* provides a range of Java APIs to access Microsoft format files based upon Microsoft's OLE 2 compound document format using pure Java (Jakarta POI, 2005). The spreadsheet generation



JavaBean is then called by PHP script to transform the product life cycle data into Excel spreadsheet, as shown in Figure 7.6.

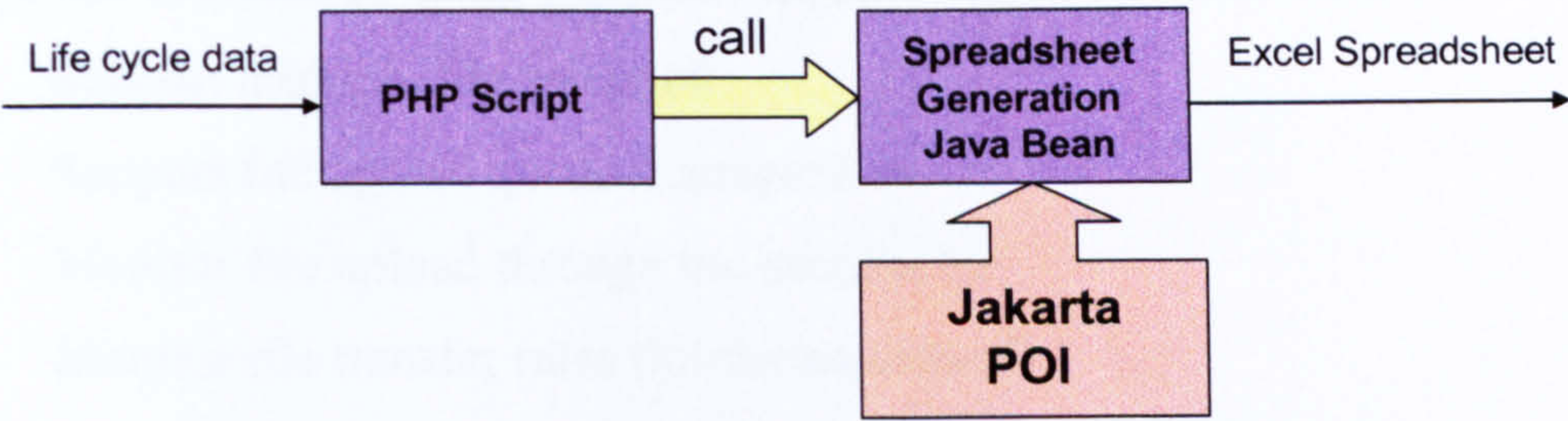


Figure 7.6 Excel spreadsheet generation

This JavaBean generating Excel spreadsheet can be found in Appendix A. (See A.1.2 *ExcelGenerator*)

7.3.4 File Upload Applet

HTML does provide a mechanism for developing file upload interface by adding ENCTYPE attribute in a HTML form as follows (Castro, 2000).

```
< Form Method = post, action="upload script" ENCTYPE="multiple/form-data" >
</Form>
```

File to upload:

Figure 7.7 Screenshot of HTML file upload interface

The user interface of this implementation for file upload is shown in Figure 7.7. However, the main disadvantage of this mechanism is that it does not support multiple file selection. For uploading one file, we have to go through these steps: (i) click the directory browse button, (ii) select one file, and (iii) click on OK. If there are 100 files to be uploaded, then we have to repeat these steps 100 times.



In order to solve this problem, an off-the-shelf component called *JUpload* (JUpload Applet, 2004) Applet is used. Figure 7.8 is a screenshot of using *JUpload* Applet for file upload. Benefits of using *JUpload* Applet are introduced as follows:

- Support multiple file selection
- Support file upload queue management
- Monitor file upload through the process bar
- Monitor file transfer rates (kilobytes/second)

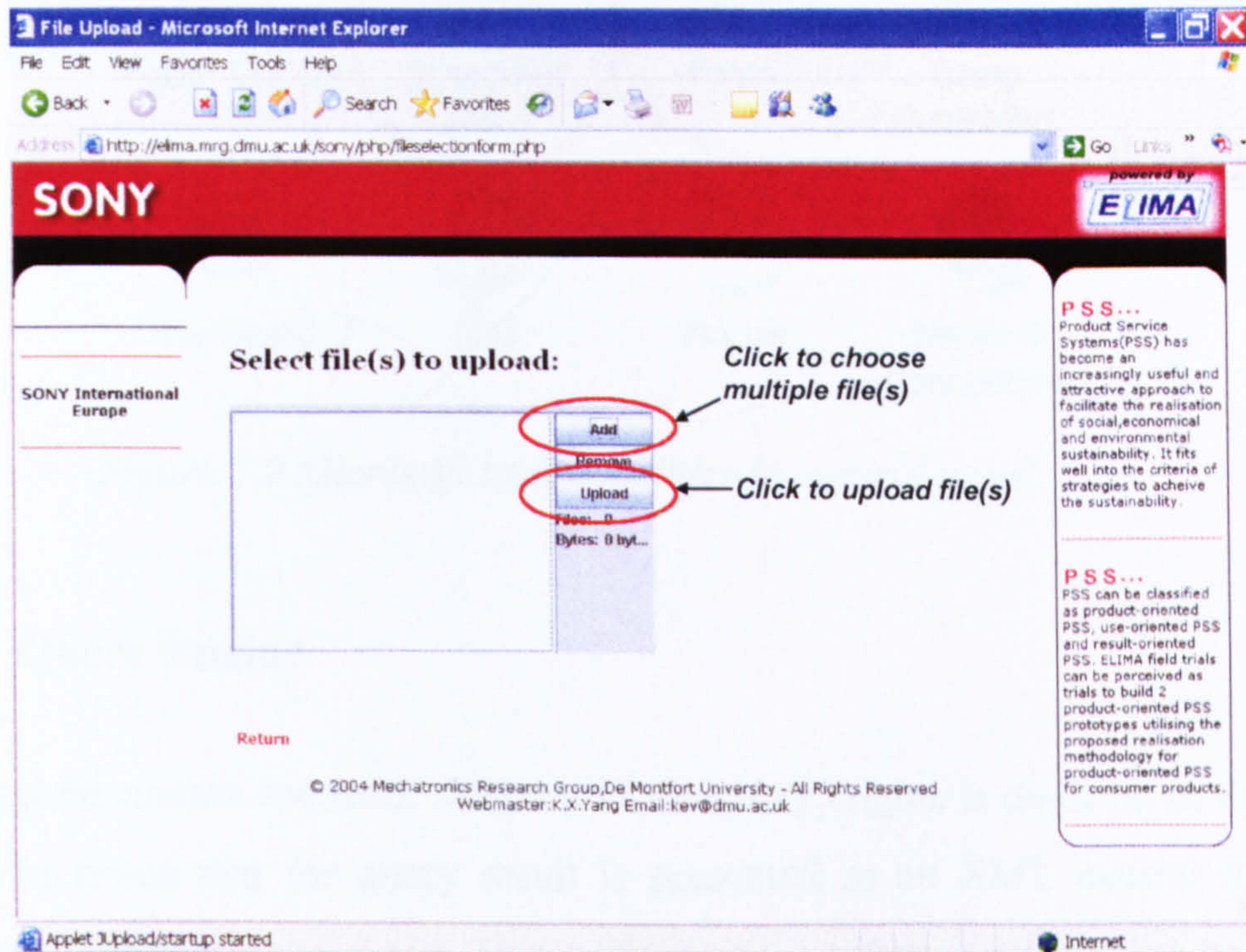


Figure 7.8 A screenshot of using *JUpload* Applet for file upload

### 7.3.5 Scheduler

The *scheduler* component in the Framework is implemented by using *Microsoft Task Scheduler*, which is shown in Figure 7.9.

The *Microsoft Task Scheduler* provides a friendly user interface which is fully integrated into the operating system, and is accessible from the *Control Panel*. Users can drag-and-drop programs right into the scheduler to quickly add a new task or use the provided "Create Scheduled Task" wizard. With *Microsoft Task Scheduler*, one can schedule any script, program, or document to be invoked at any time or any interval,



every day to once a year, and on events like system boot, user logon, or system idle; a task is saved as a file with a .job extension, which enhances the ability to move from computer to computer (Microsoft Task Scheduler, 2005).

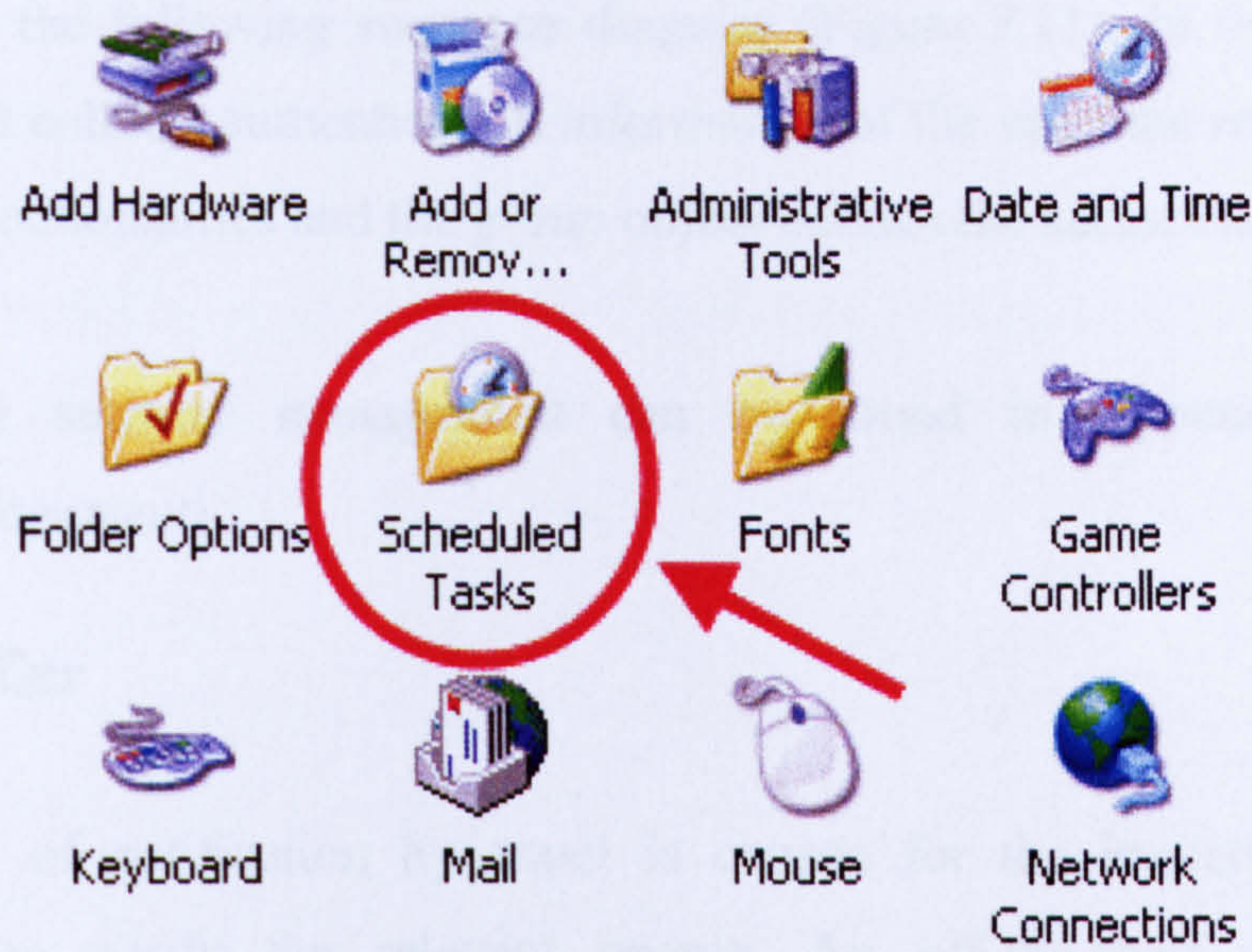


Figure 7.9 Microsoft task scheduler in control panel

7.3.6 Query Engine

The implementation sequence diagram of the query engine is depicted in Figure 7.10. It has to be noted that the query result is presented in an XML neutral format before passing to the report generator. This enables other software application or organisation to utilise the data with ease.

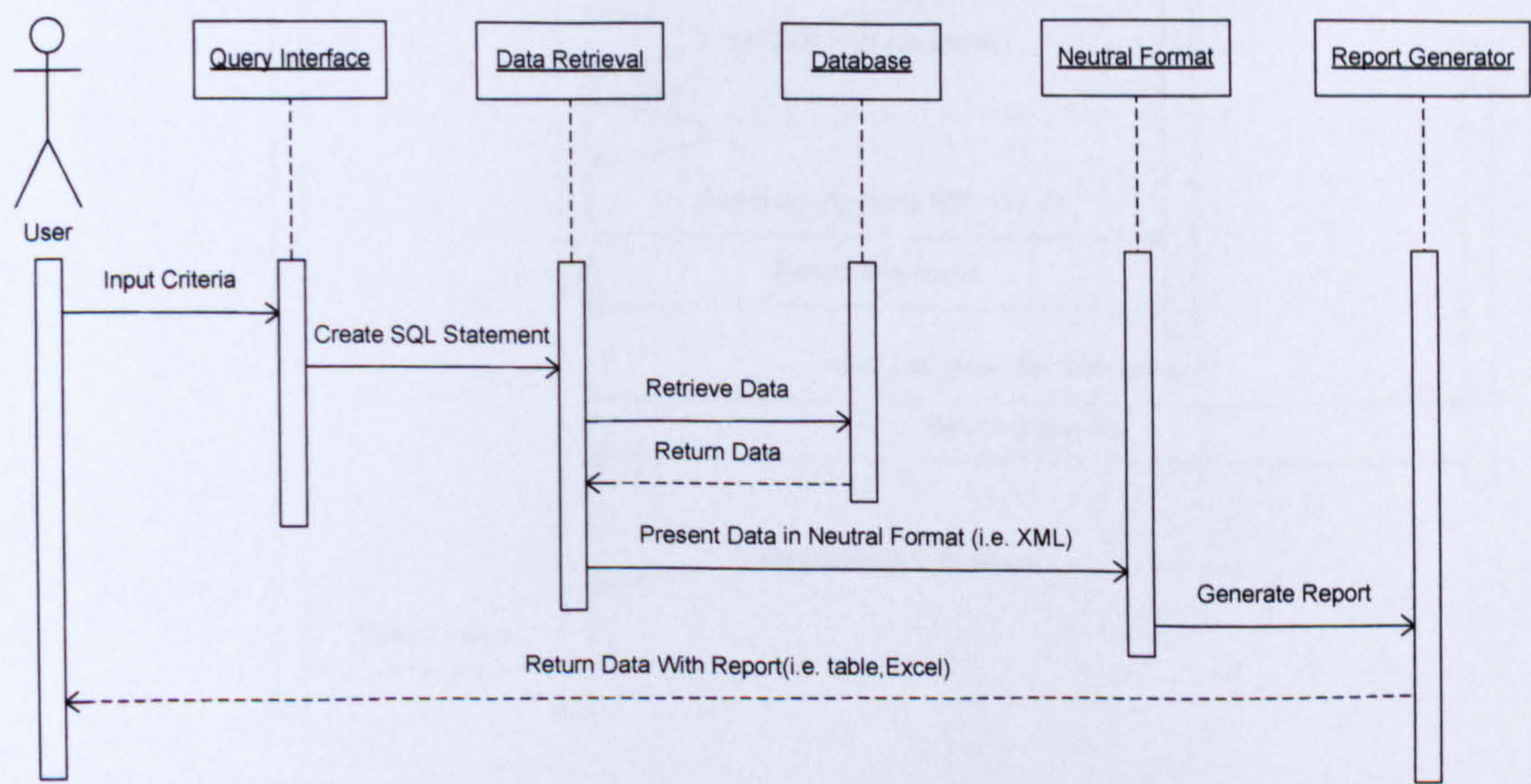


Figure 7.10 Sequence diagram of query operation



7.3.7 Security Manager

The implementation of the *three-level access control* and *role-session* paradigm is explained in the following sequence diagram (Figure 7.11). In the diagram, the *user profile* object collects authentication information of the user; the *role* object defines the accessible functionalities and the *group* object defines the accessible dynamic data.

A class for security management can be found in Appendix A. (See A.2.3 *SecurityManagement*)

7.3.8 Notifier

The method of notification by email is chosen for the implementation of *notifier* component to notify the relevant people. An off-the-shelf component, namely, *JavaMail* is employed to implement the *notifier*. The *JavaMail* API provides a platform-independent and protocol-independent framework to build mail and messaging applications (Java Mail, 2005).

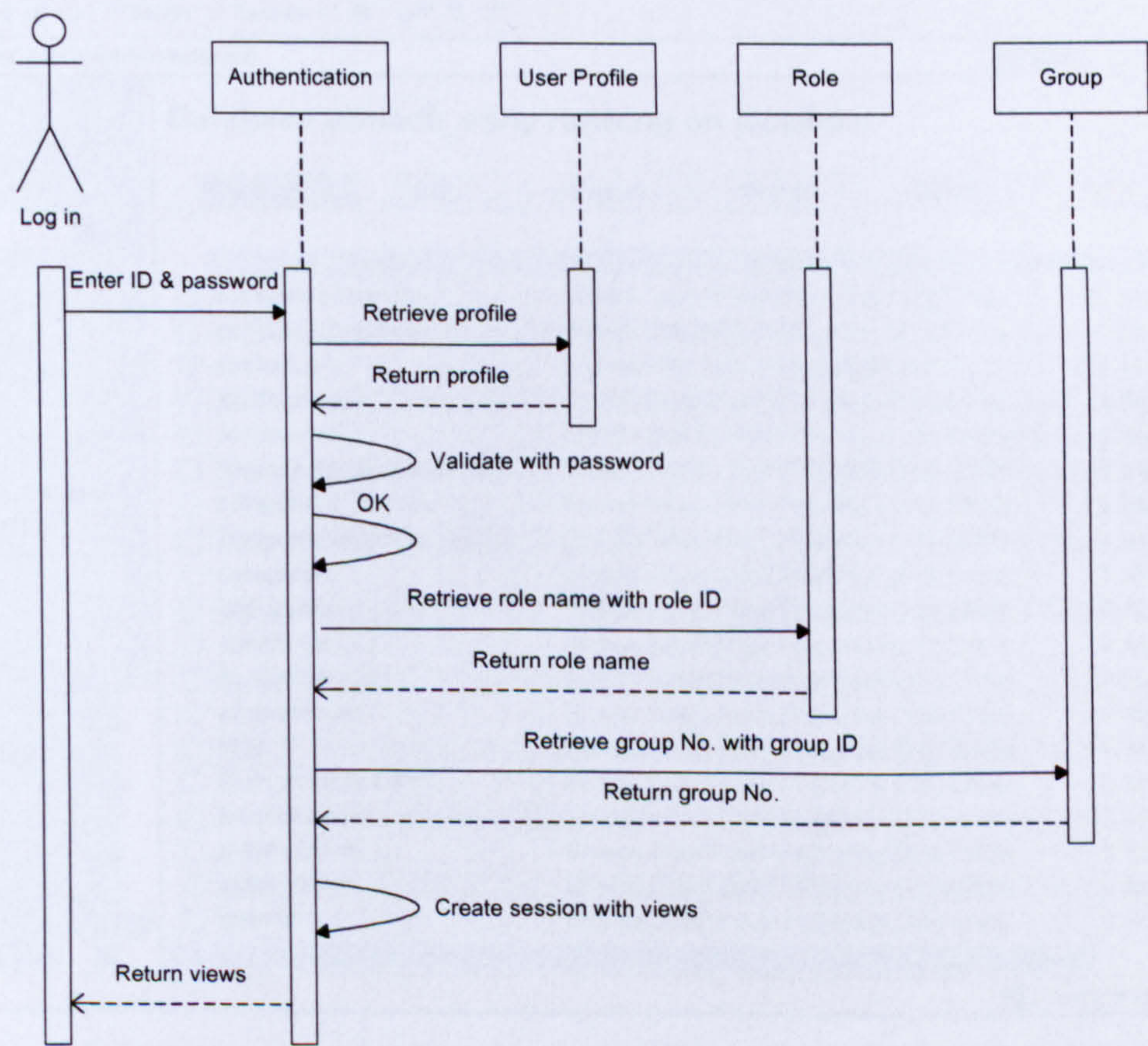


Figure 7.11 Operation sequence of security model



7.3.9 Database Administration

The *database administration* component is implemented by adopting the off-the-shelf component *PhpMyAdmin* (PhpMyAdmin, 2005). *PhpMyAdmin* is a tool written in PHP intended to handle the administration of *MySQL* DBMS over the Web. The *PhpMyAdmin 2.3.2* can create and drop databases, create/drop/alter tables, delete/edit/add fields, execute any SQL statement, manage privileges, export data into various formats. A screenshot of the incorporating *PhpMyAdmin 2.3.2* is shown in Figure 7.12.

The *PhpMyAdmin* provides an intuitive way for database administration. For example, if any box is ticked, the corresponding menu such as *Browse*, *Select*, etc is activated for required operation. In Figure 7.12, the table *appliance* is chosen. If we click *Browse*, all the data in *appliance* will be displayed.

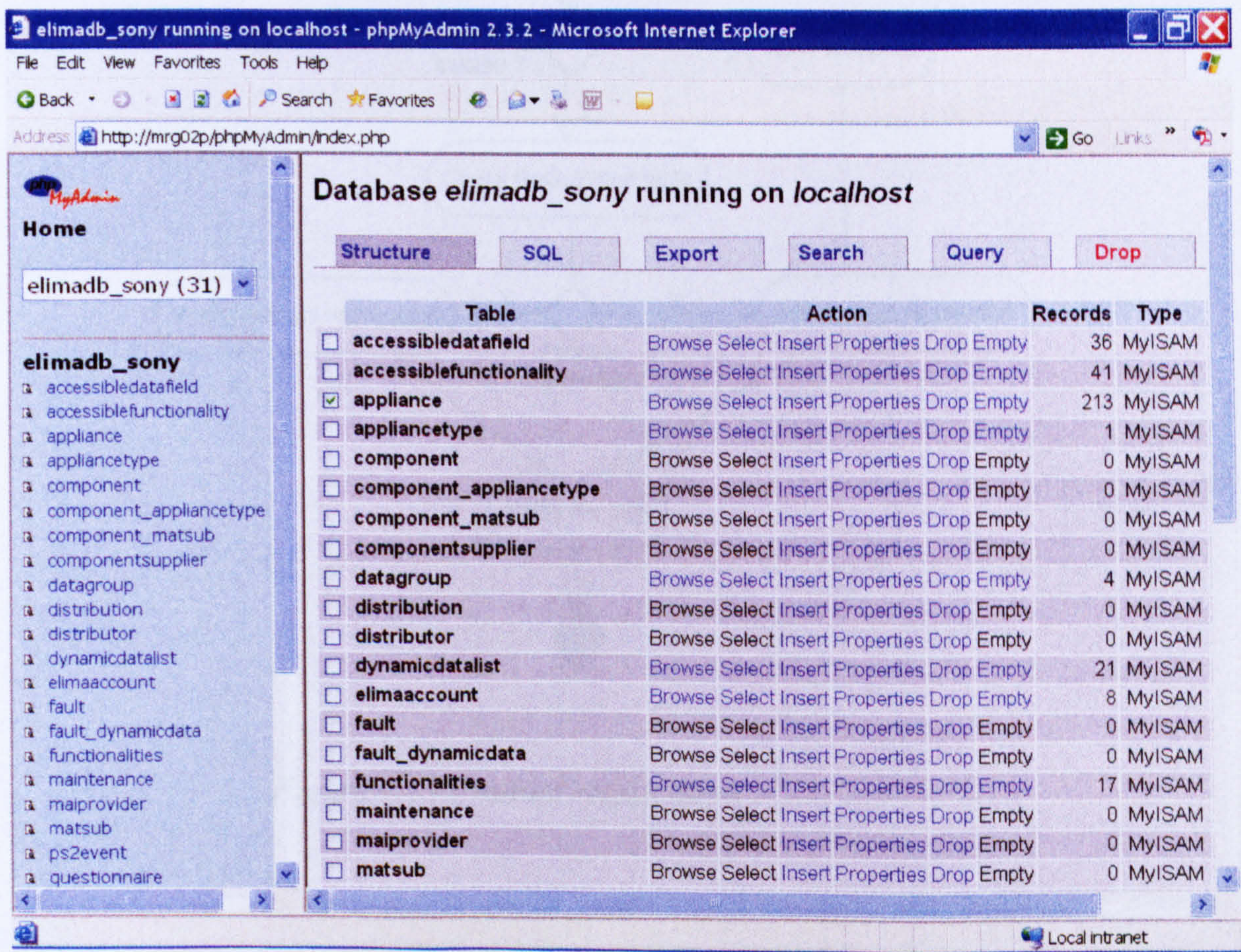


Figure 7.12 A screenshot of PhpMyAdmin



7.3.10 Product Registration

The implementation activity diagram of the *product registration* is shown in Figure 7.13. The data collected through product registration is stored centrally in the database, and linked dynamically to the dynamic data of the product. When the product is registered, the *notifier* will notify the product user that the registration is successful. If the product is not registered, some of the operation may be disabled. The reason is that some operations need product registration data. For example, calculation of some of the extended dataset such as *average operation time*, *product usage duration*, etc may require the product usage start date to be acquired.

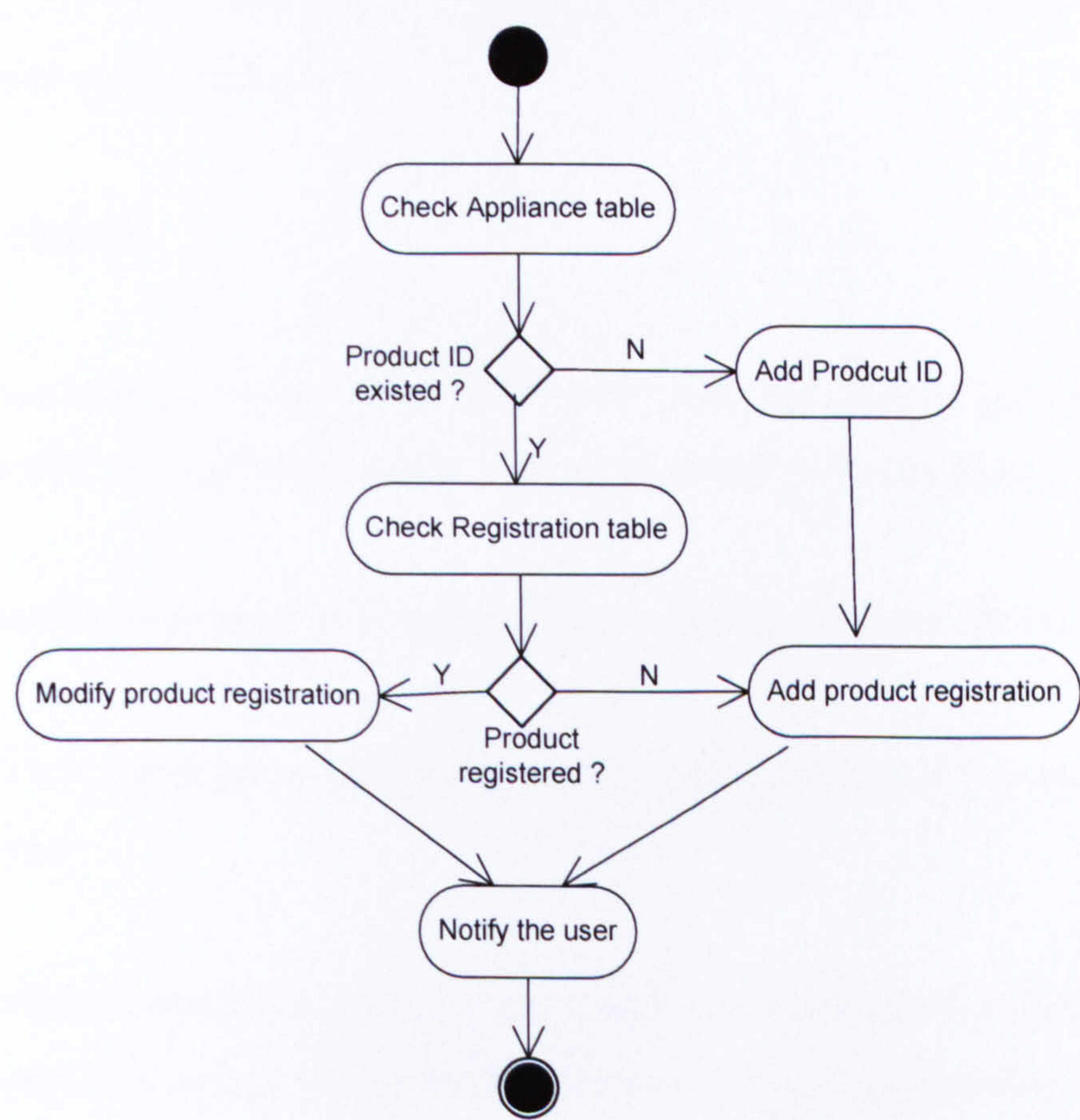


Figure 7.13 Product registration



## 7.4 Supporting Tools

Apart from the development of the Framework, some associated supporting tools are also developed to facilitate the creation of product-specific service enabler. These supporting tools include:

- Database management supporting tools
- Service enabler creation tools
- Useful utilities.

Implementations of these supporting tools involve employment of class libraries and APIs provided by the Framework. The description of some supporting tools formulated can be found in Appendix B.

## 7.5 Summary

The implementation of the Framework and associated supporting tools for realising service enabler for consumer products were discussed in this chapter:

- 1) The development tools and implementation strategy was introduced.
- 2) The implementation of underlying runtime infrastructure of the service enabler was described.
- 3) The implementation of each identified generic component has been described. The utilisation of standard off-the-shelf components was demonstrated.



# Chapter 8 Using the Framework to Realise a Service Enabler Solution

## 8.1 Introduction

This chapter describes how the Framework can be used to create a product-specific service enabler. Usually software framework provides plug points for developers to customise the framework. Plug points are pre-defined places where developers can focus on specific business logic programming (Short 1997; Moore et al, 2003). In this Chapter, Section 8.2 introduces the plug points provided by the Framework, while Section 8.3 describes further details about the creation process.

The process of using the Framework to create a product-specific service enabler is illustrated in Figure 8.1.

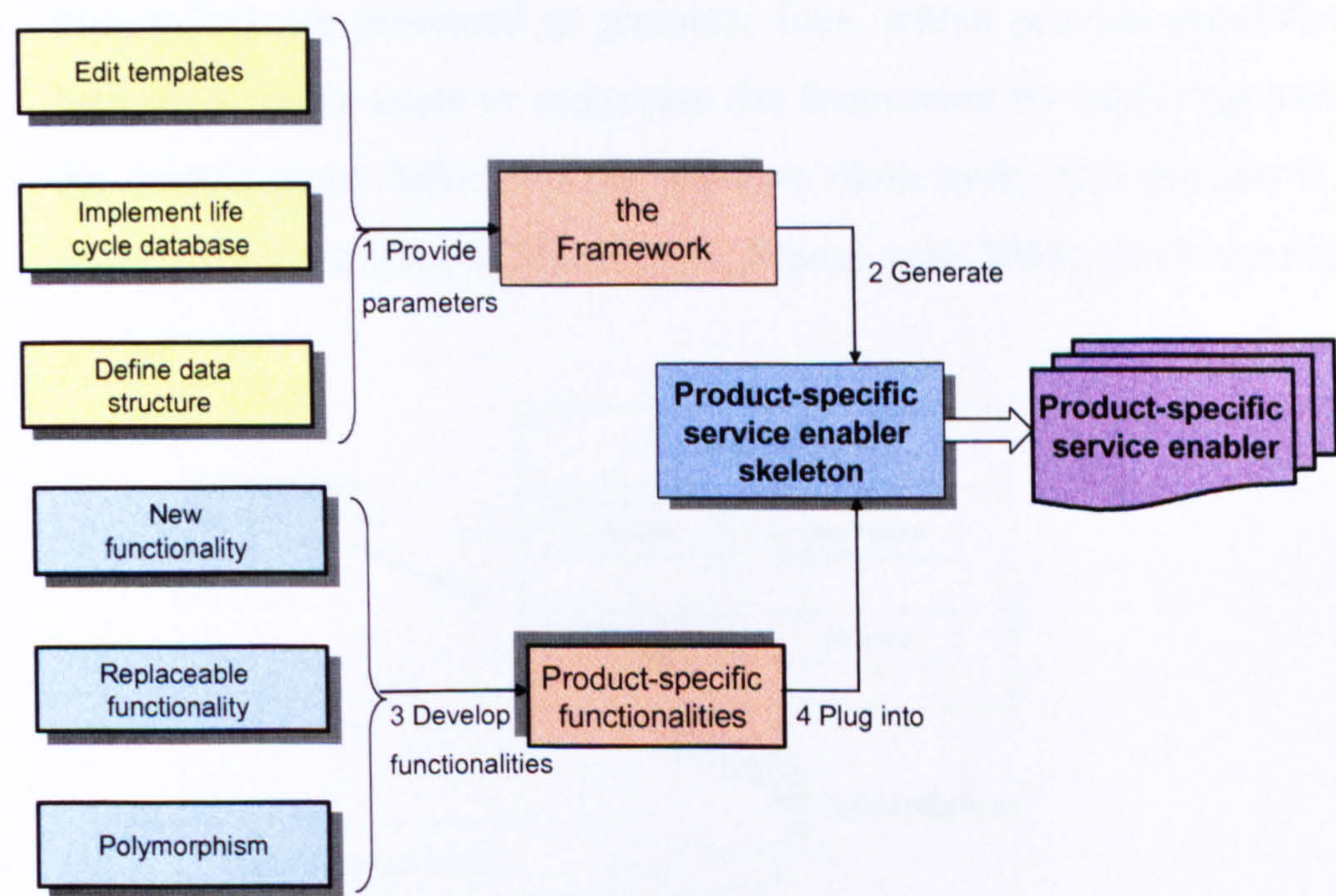


Figure 8.1 The process of using the Framework to realise a service enabler solution



## 8.2 Plug Points

The Framework provides plug points for developers to concentrate on product-specific business logic programming in the following three forms.

- Plug points are presented as specifications of product-specific components (e.g. communication manager). Developers are required to provide code to the product-specific component based on the specification of relevant interfaces. This can be regarded as the provided code “fills in the blanks” left deliberately by the software framework (Short 1997).
- Plug points may be presented as some default generic components or functionalities. Developers can replace them with product-specific components or functionalities.
- Plug points are presented as template files, which provide pre-defined design patterns to guide users to customise the framework by supplying parameters to the generic terms defined in the software framework. This process is known as framework instantiation (Short, 1997; Moore et al, 2003), as shown Figure 8.2.

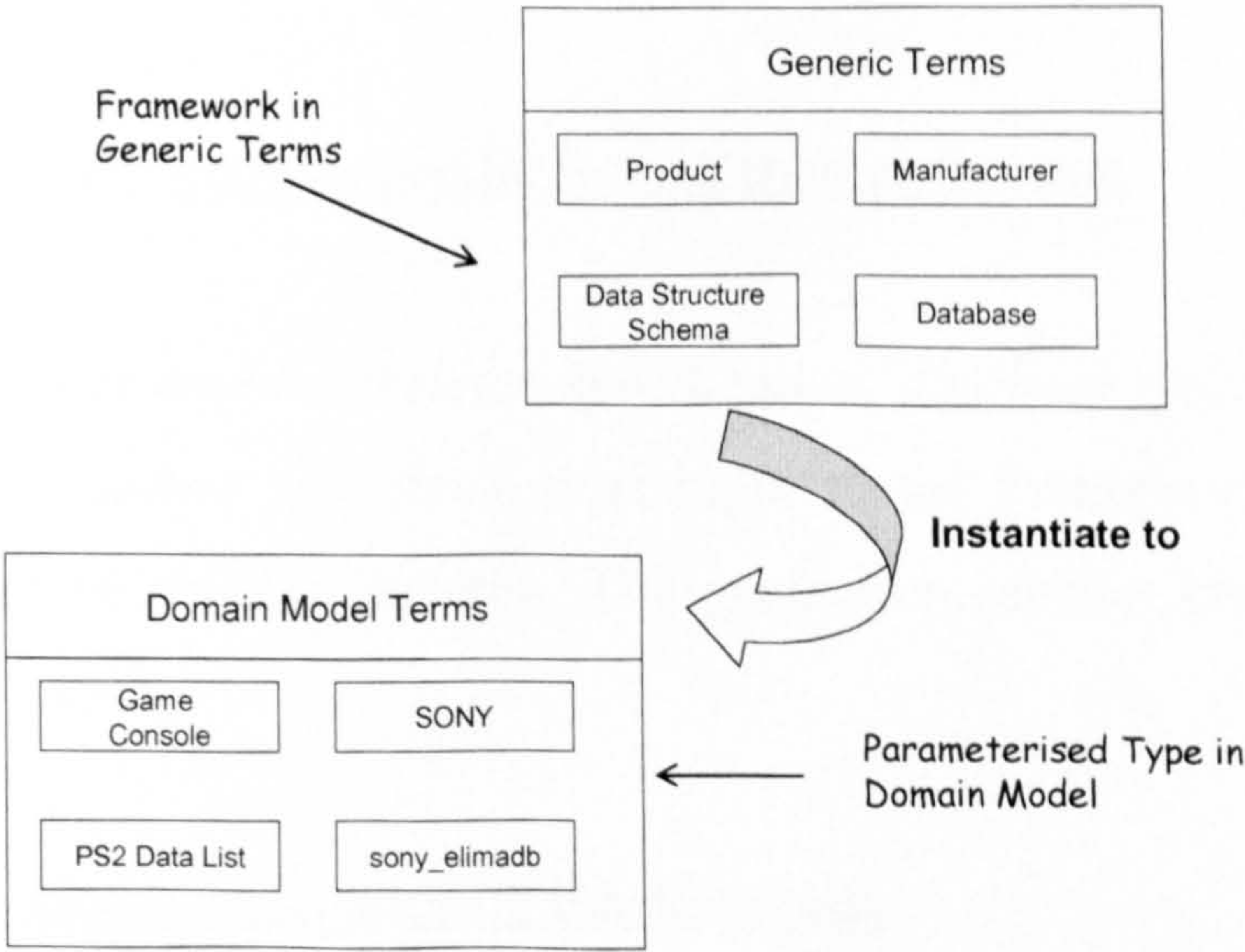


Figure 8.2 The software framework instantiation



### 8.3 Service Enabler Realisation Process

The process of using the Framework to create a product-specific service enabler is described as follows, as shown in Figure 8.2.

#### Step 1: Provide product-specific parameters at plug points

- Based on the generic template files (i.e. Java based template file and PHP based template file) provided by the Framework, fill in the product-specific primary data fields (e.g. product name, dynamic data), extended data fields, concrete parameters, specific handlers, or values, etc to formulate a *product-specific service enabler skeleton specification* (i.e. script).
- Based on the generic database schema provided by the Framework, define product-specific primary data fields and product-specific extended data fields, etc to formulate a *product-specific database specification* (i.e. MySql script).
- Based on the generic dynamic life cycle data structure schema provided by the Framework, define the product-specific data type, event type, etc to formulate a *product-specific dynamic life cycle data structure specification* (i.e. XML document).

#### Step 2: Generate product-specific service enabler skeleton

Take the *service enabler skeleton specification*, *database specification* and *dynamic life cycle data structure specification* as input to the Framework to generate a product-specific service enabler skeleton. The generation process can be automated by using supporting tools

#### Step 3: Develop product-specific functionalities

Based on interfaces provided by the Framework, product-specific functionalities can be developed. This development can be any of the following ways:



- Develop new product-specific functionality to plug into the service enabler skeleton.
- Develop functionality to replace Framework-provided default functionality.
- Polymorphism. Implement the product-specific functionality by incorporating the methods and signatures declared by interfaces. The functionality will be auto-embedded into the service enabler skeleton via OO inheritance and polymorphism mechanism.

#### Step 4 Plug specific functionalities into the skeleton

Plug developed product-specific functionalities into the service enabler skeleton to form a required service enabler. The plugging process can be achieved with the aid of supporting tools.

### **8.4 Summary**

Using the Framework to create a product-specific service enabler has been described in this chapter:

- 1) Plug points are pre-defined places where developers can focus on the domain-specific business logic programming.
- 2) A process of using Framework to create a product-specific service enabler was described. The realisation process is through 4 steps:
  - (i) Provide product-specific parameters at plug points.
  - (ii) Generate product-specific service enabler skeleton.
  - (iii) Develop product-specific functionalities.
  - (iv) Plug product-specific functionalities into the skeleton to eventually form a product-specific service enabler.



## Chapter 9 Test Cases

### 9.1 Introduction

This chapter introduces two test cases which are designed to test the Framework and the proposed PSS realisation methodology. The test approach can be described as:

- Build product-oriented PSS prototype using the proposed methodology and the Framework.
- Use the prototype to test the service enabler and to demonstrate that the expected services are enabled.
- Evaluate test results.

The test cases adopt two product-oriented PSS prototypes built in ELIMA project. Two specific service enablers, namely, PSIMS and FFIMS created by using the Framework are introduced. Enabled services from product life cycle data via PSIMS and FFIMS are demonstrated.

This chapter is organised as follows:

- Section 9.2 introduces the test case design.
- Section 9.3 discusses the test case of fridge/freezer PSS prototype.
- Section 9.4 discusses the test case of PS2 PSS prototype.
- Section 9.5 appraises the test results.

### 9.2 Test Case Design

Test case typically describes actions or steps and the expected results due to the actions or steps (Sisson, 2002). Test case design includes the test scope, test approach, expected results as well as assumptions and constraints.



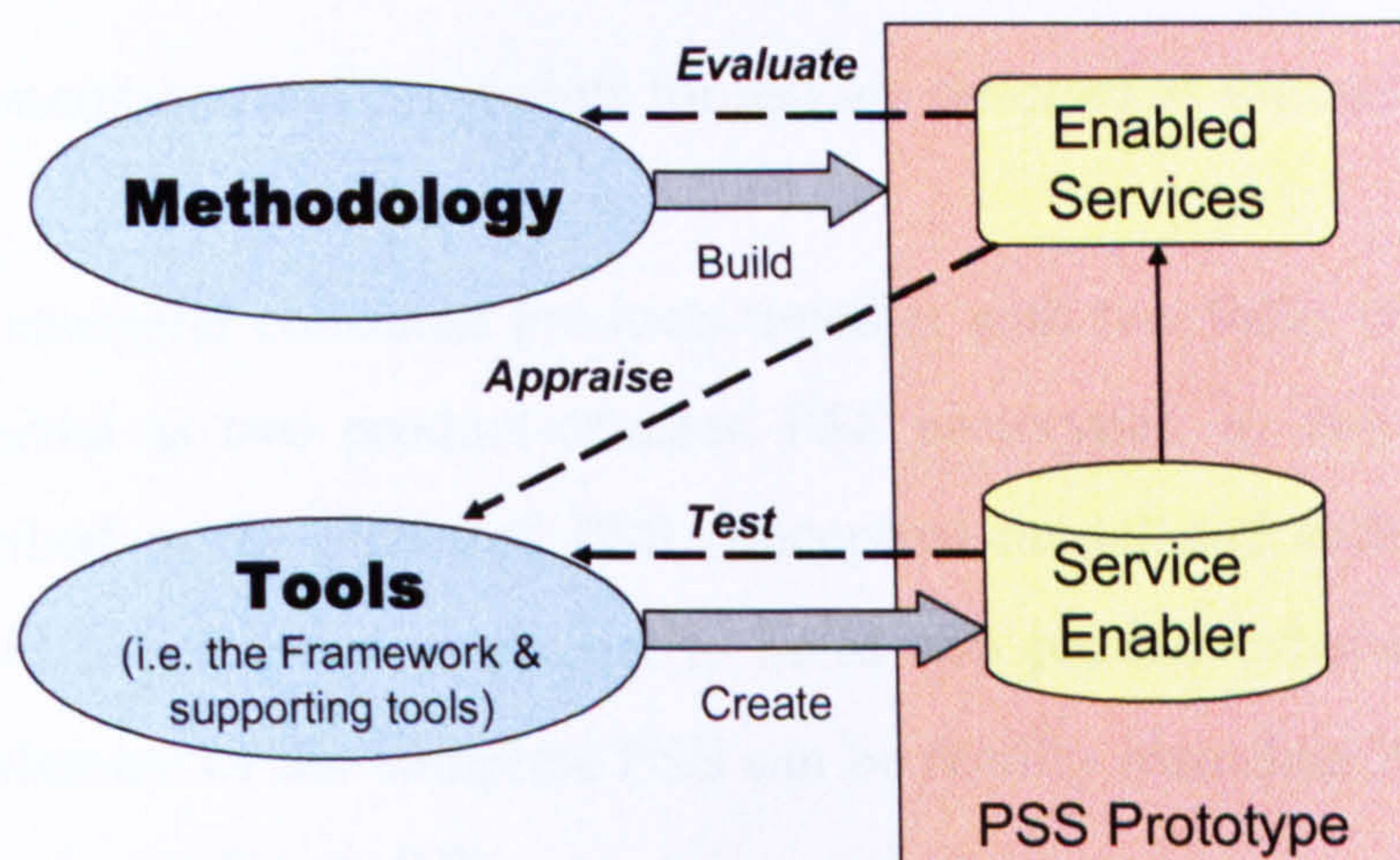
### 9.2.1 Test Scope

The test scope is introduced as follows:

- Test the appropriateness and effectiveness of the overall proposed methodology for the realisation of PSS for consumer products.
- Test the effectiveness the Framework and associated supporting tools which facilitate the creation of the service enabler.
- Demonstrate the enabled services which are delivered via PSS prototypes.

### 9.2.2 Test Approach

The test approach is shown in Figure 9.1:



*Figure 9.1 Test approach*

- 1) Using the proposed PSS realisation methodology and the Framework to build two product-oriented PSS prototypes.
- 2) Using the developed service enabler to appraise the Framework.
- 3) Using the PSS prototype to demonstrate the enabled services so that the proposed methodology and the Framework can be evaluated.
- 4) Test results discussion. This includes the comparison of the test results against the expected results and the realisation of sustainability.



9.2.3 Expected Results and Test Criteria

The expected result is discussed as follows:

- 1) In the prototype PSS:
  - the product life cycle data should be acquired
  - the acquired life cycle data should be transmitted to the service enabler
  - service enabler can show enabling services
- 2) Product-specific service enabler can be created by using the Framework, and the developed service enabler should meet the requirements specification.
- 3) PSS prototype should help the achievement of the sustainability.

9.2.4 Assumptions and Constraints

Some assumptions and constraints for test are described as follows:

- Two exemplar consumer products together with two IMSs in ELIMA project can be perceived as two product-oriented PSS prototypes, as they meet the PSS criteria described in the proposed PSS conceptual model and architecture. Thus, ELIMA project is perceived as project to build two product-oriented PSS prototypes. The key element of the complete PSS can be readily identified in the ELIMA project as shown in the Figure 9.2.

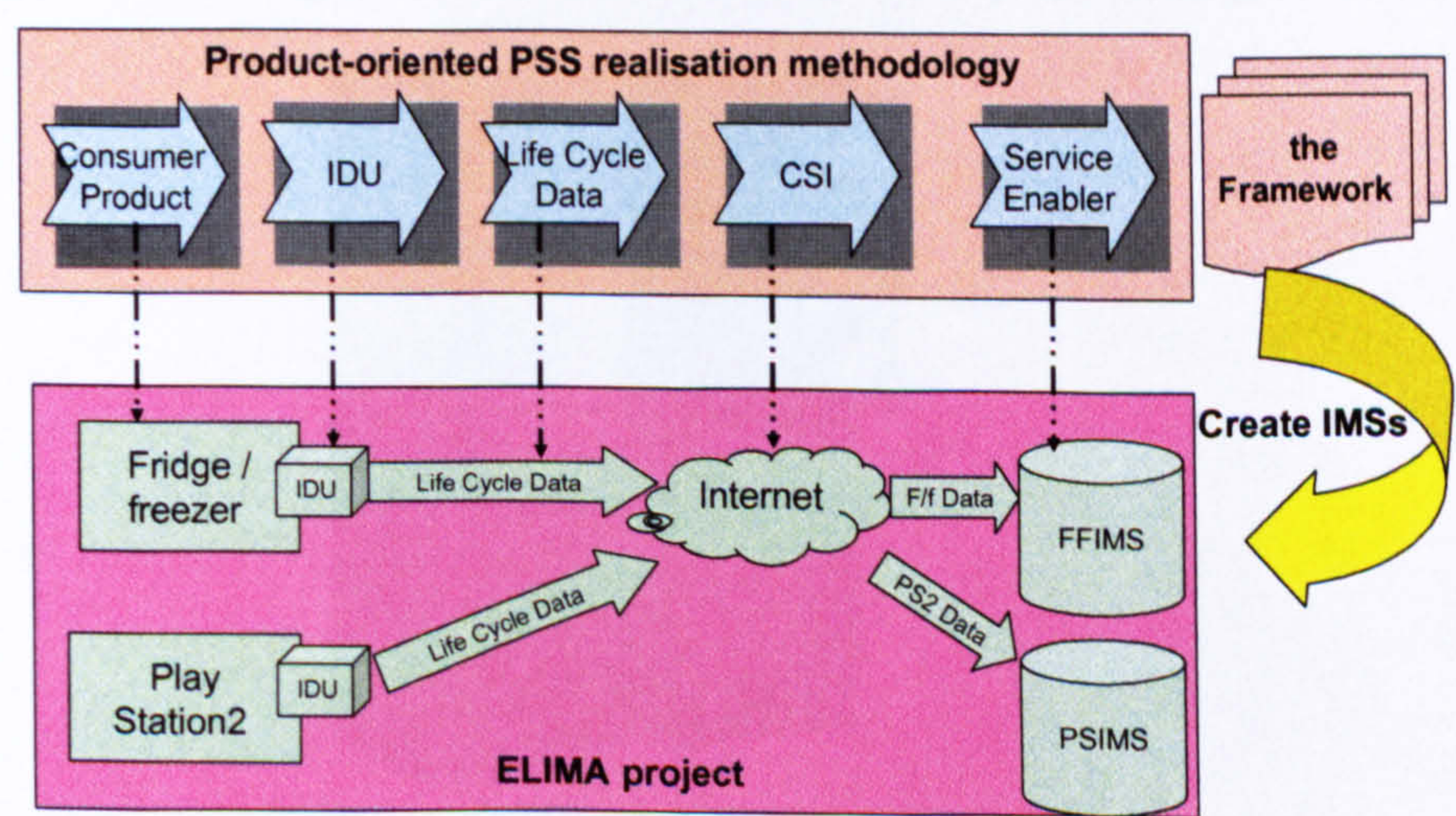


Figure 9.2 Product-oriented PSS prototypes built in ELIMA project



- All the data acquired by IDU is assumed to be good quality data.
- Data privacy is not the main issue.
- Statements such as “*services enabled from life cycle data*”, “*services enabled from service enabler*” and “*services delivered from PSS*”, etc have the same meaning.

### 9.3 Test Case I: Fridge/freezer PSS Prototype

In this test case, we will:

- Use the Framework to create FFIMS and integrate it into the trial.
- Evaluate the FFIMS.
- Demonstrate services which are delivered via the PSS prototype.

#### 9.3.1 ELIMA White Goods Field Trial Overview

The white goods field trial is undertaken by Merloni UK, which is one of the UK’s largest manufacturers of domestic appliances. They intend to test the ELIMA concept on white goods, looking for benefits in maintenance, reliability prediction and home automation (ELIMA, 2005).

##### 9.3.1.1 Trial Product and Specification

The fridge/freezer (Figure 9.3) is chosen as an exemplar white goods for the field trial.



*Figure 9.3 Trial product: fridge/freezer*



The white goods field trial includes 8 fridge/freezer product models. The trial generates data on three levels (ELIMA D17, 2004):

- Level 1: Standard Merloni UK field trial – where questionnaires are sent to field trial machine owners. At this level products are distributed to Merloni employees and employees' families and friends.

In this level, the following data is collected through the questionnaire.

- Installation date
  - Packaging information
  - Appliance instruction
  - Noise levels
  - Lighting
  - Defrosting
  - Cleaning
  - Rank value of functions and features
  - Adjusting temperature
  - Appearance
  - Any problems
- Level 2: Statistical dynamic data collection – where an EEPROM is an integral part of the appliances' electronic control module. A 256 byte EEPROM is chosen to provide adequate capacity to store the acquired life cycle data. Data is acquired by visiting customer's homes and downloading statistical data to a laptop PC from the integrated IDU.

In this level, the following data can be obtained from the IDU.

- Compressor run time
- Ambient temperature °C
- Door open counts
- Cumulative door open time
- Defrost time



- Number of compressor on/off times
  - Cumulative compressor run time
  - Number of power on/off
  - Number of high temp alarms
  - Number of frost free cycles
  - Number of power cuts
  - Number of times the holiday mode is selected
  - Number of times the sensor fault is detected
  - Number of valve cycles
- Level 3: Dynamic GSM data with data logging capability – remote real time data logging of energy, temperature and statistical data via a purpose built IDU. Six products are fitted with a remote logging hardware that transmits data using GSM<sup>5</sup>.

At this level, the following data is collected:

- Door open events: Door data when either fridge or freezer door is opened
- Energy: Cumulative energy every minute and every three hours
- Temperature: Logged every minute from different kind of temperature sensors
- Statistical data: EEPROM data collected every twenty-four hours
- Voltage: Logged every three hours
- Current: Logged every three hours

### 9.3.1.2 IDU and Data Acquisition

There are two mechanisms for life cycle data acquisition through IDU in white goods field trial: (i) data download from EEPROM and (ii) data transmission through GSM.

- Data download from EEPROM

In this mechanism, a range of sensors (i.e. temperature sensor, humidity sensor) and an EEPROM are embedded into the fridge/freezer to act as an IDU. The life cycle data

---

<sup>5</sup> As the project progresses, it is found that the software could be easily modified to make it work over a normal phone line. Hence a PSTN modem based product have also been designed and installed.



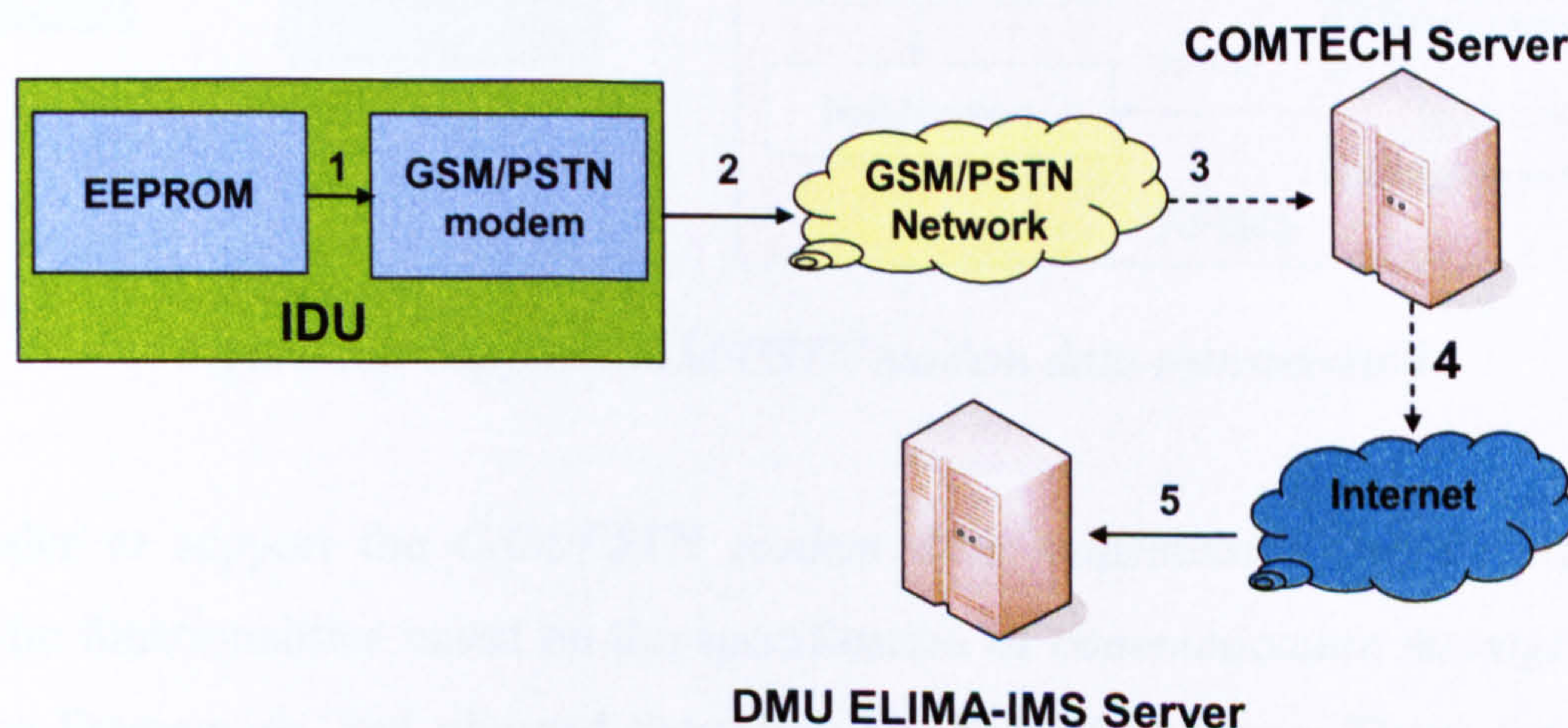
captured by sensors is stored in the EEPROM. An engineer has to visit customer's house and download data from EEPROM to a laptop to acquire the data. The data is normally saved as a 2 kb CSV text file. The data is linked to the product serial number and includes the date and time of download. The raw text format consists of two columns, the first column contains the EEPROM address and the second column contains the counts and the operational settings. The FFIMS provides the functionality to allow generated text file to be uploaded to the FFIMS database over the Internet.

- Data transmission through GSM/PSTN modem

In this mechanism, the IDU comprises:

- A module to collect data from the appliance's EEPROM
- An accurate electricity-consumption meter
- A GSM mobile telephone modem/PSTN modem

Data in EEPROM and energy consumption data are transferred to a secure server hosted by Comtech<sup>6</sup> via the GSM/PSTN modem and corresponding networks. Comtech server then automatically transmits the data to DMU FFIMS server at 24-hour intervals in a form of file. Figure 9.4 illustrates this type of data transmission.



*Figure 9.4 Data transmission through GSM/PSTN modem*

<sup>6</sup> COMTECH is the contractor responsible for collecting data through GSM modem from the fridge/freezer before repackaging the data and sending them to DMU FFIMS



9.3.2 Using the Framework to Create FFIMS

The FFIMS skeleton is created by using the Framework. In order to support the specific features of white goods field trial such as GSM/PSTN data transmission, etc, specific developments are described as follows.

Life cycle data transfer in live manner has been demonstrated in white goods field trial. From the perspective of COMTEC server, *GSM/PSTN modem data transmission* is a live data transfer. But in the real world in future, the life cycle data will be directly transmitted to the service enabler server lively.

In the trial, from the perspective of DMU FFIMS server, data transmission is still the file transfer. But this type of file transfer can not use the manually file upload mechanism as the file is automatically scheduled to be transmitted to DMU server every 24 hours.

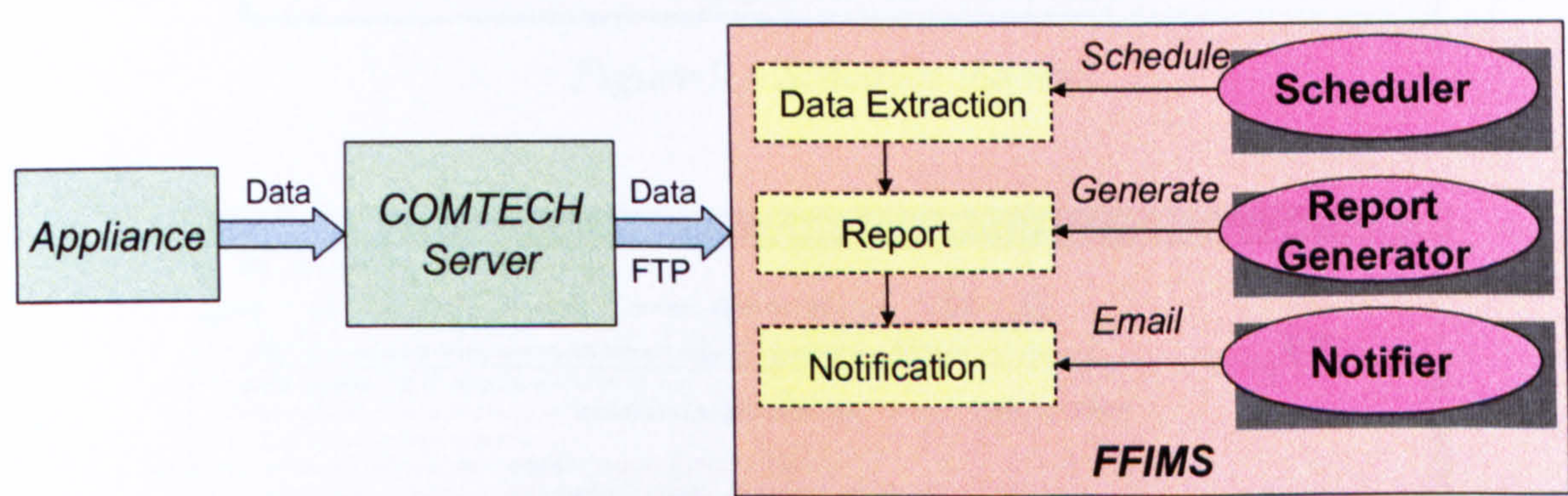


Figure 9.5 Support GSM/PSTN modem data transmission

In order to support the *GSM/PSTN modem data transmission*, we developed some specific functionalities based on the specification of *communication manager* provided by the Framework, and plugged them into the FFIMS skeleton. These functionalities work in corporation with the *scheduler*, *report generator* and *notifier* to fulfil the required jobs (shown in Figure 9.5). In field trial, the COMTECH server is scheduled to transmit the packaged data (i.e. file) to FFIMS automatically at 1:00 am every morning. Thus the *scheduler* of FFIMS is scheduled to perform the data extraction task at 3:00am after file being received (shown in Figure 9.6). When the data extraction task has done,



the *report generator* generates a data extraction summary report, which is presented in XML (Figure 9.7). The generated report is then automatically emailed to the relevant engineers at Merloni UK by the *notifier* immediately.

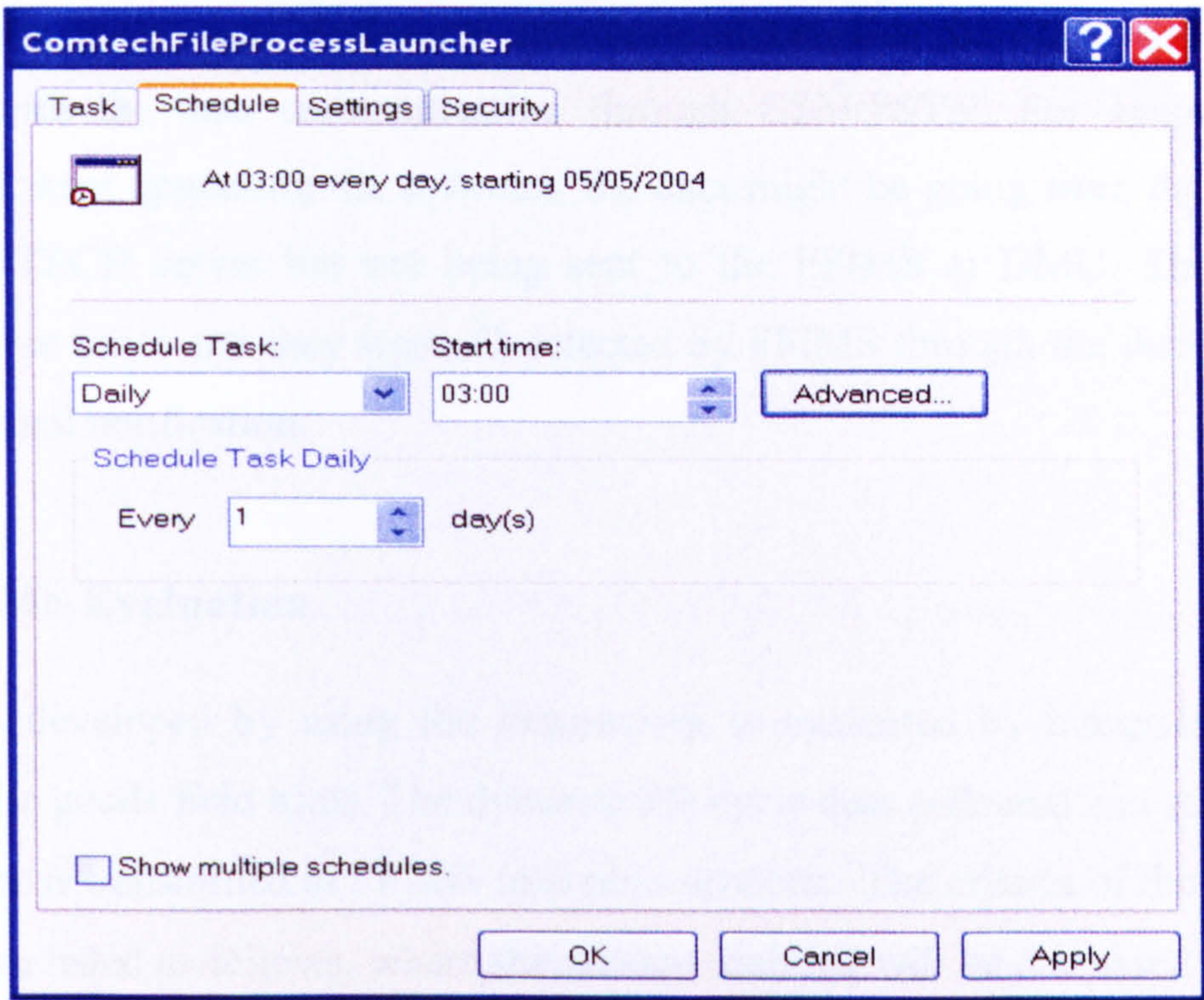


Figure 9.6 Schedule the task

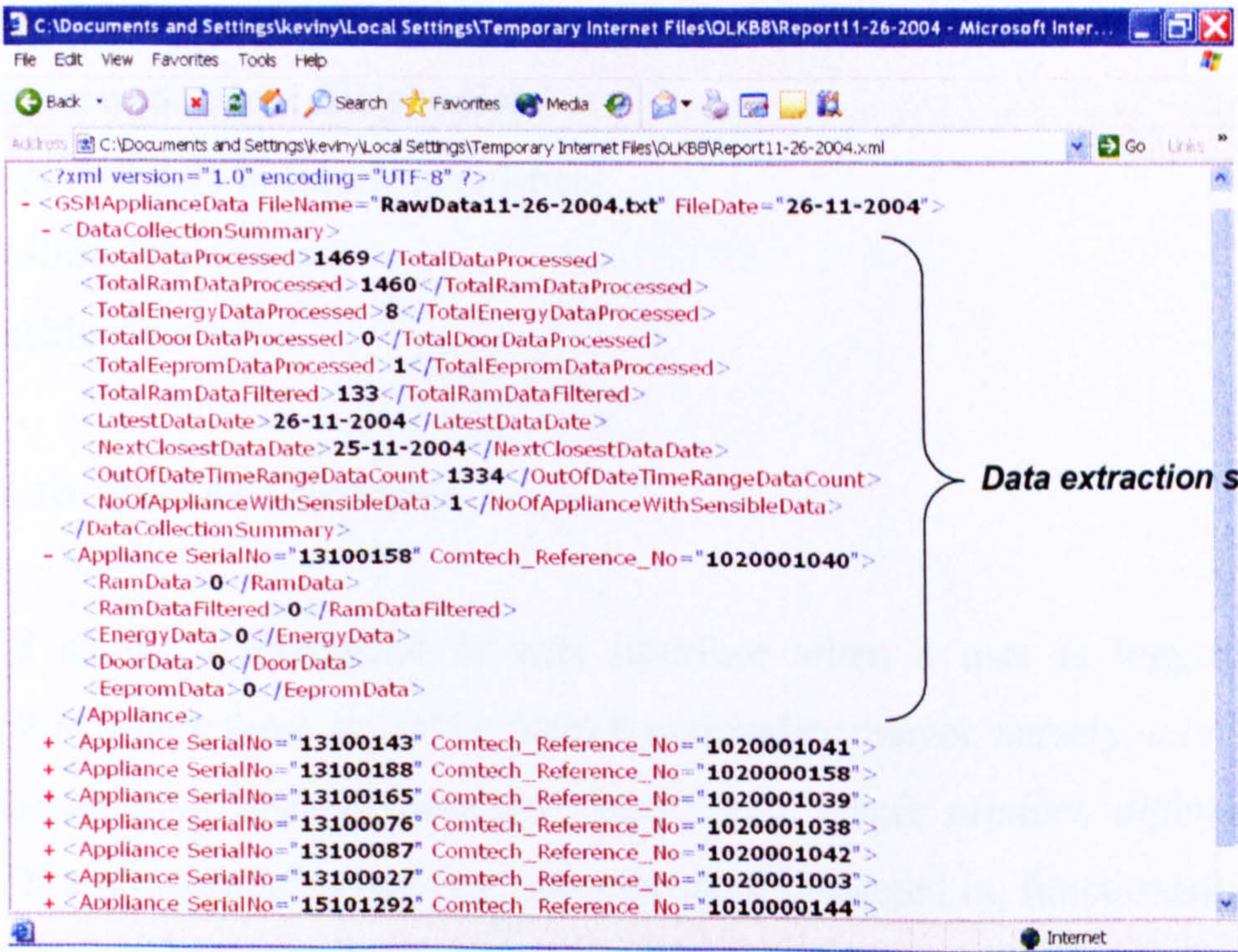


Figure 9.7 A snapshot of data extraction summary report



The white goods field trial has demonstrated that these functionalities successfully supported the *GSM/PSTN modem data transmission*. From May, 2004 to December, 2004, FFIMS received files sent from the COMTECH server almost every night. Moreover, the daily data extraction summary report also helped track problems associated with the data communication through GSM/PSTN. For example, since Comtech UK kept upgrading the software, the data might be going from the appliance to the COMTECH server but not being sent to the FFIMS at DMU. This problem happened three times and they were all detected by FFIMS through the data extraction report and email notification.

### 9.3.3 FFIMS Evaluation

The FFIMS developed by using the Framework is evaluated by integrated into the ELIMA white goods field trials. The dynamic life cycle data collected and static data of fridge/freezer is transmitted to FFIMS to enable services. The criteria of the evaluation of FFIMS are listed as follows, where the service enabling will be discussed in the next section (Section 9.3.4).

- 1) User interface
- 2) Security management
- 3) Data transmission: file(s) upload
- 4) Export of data to Excel spreadsheet
- 5) On-line data analysis
- 6) Enabling services

#### 9.3.3.1 User Interface Overview

Figure 9.8 shows a screenshot of user interface when a user is logged in as an administrator, where there are seven main functionality menus, namely, *administration*, *data communication*, *security manager*, *static data*, *single product*, *different product* and *help*. If a non-administrator (e.g. recycler) role is logged in, functionalities such as *administration* and *security manager* will not be shown.



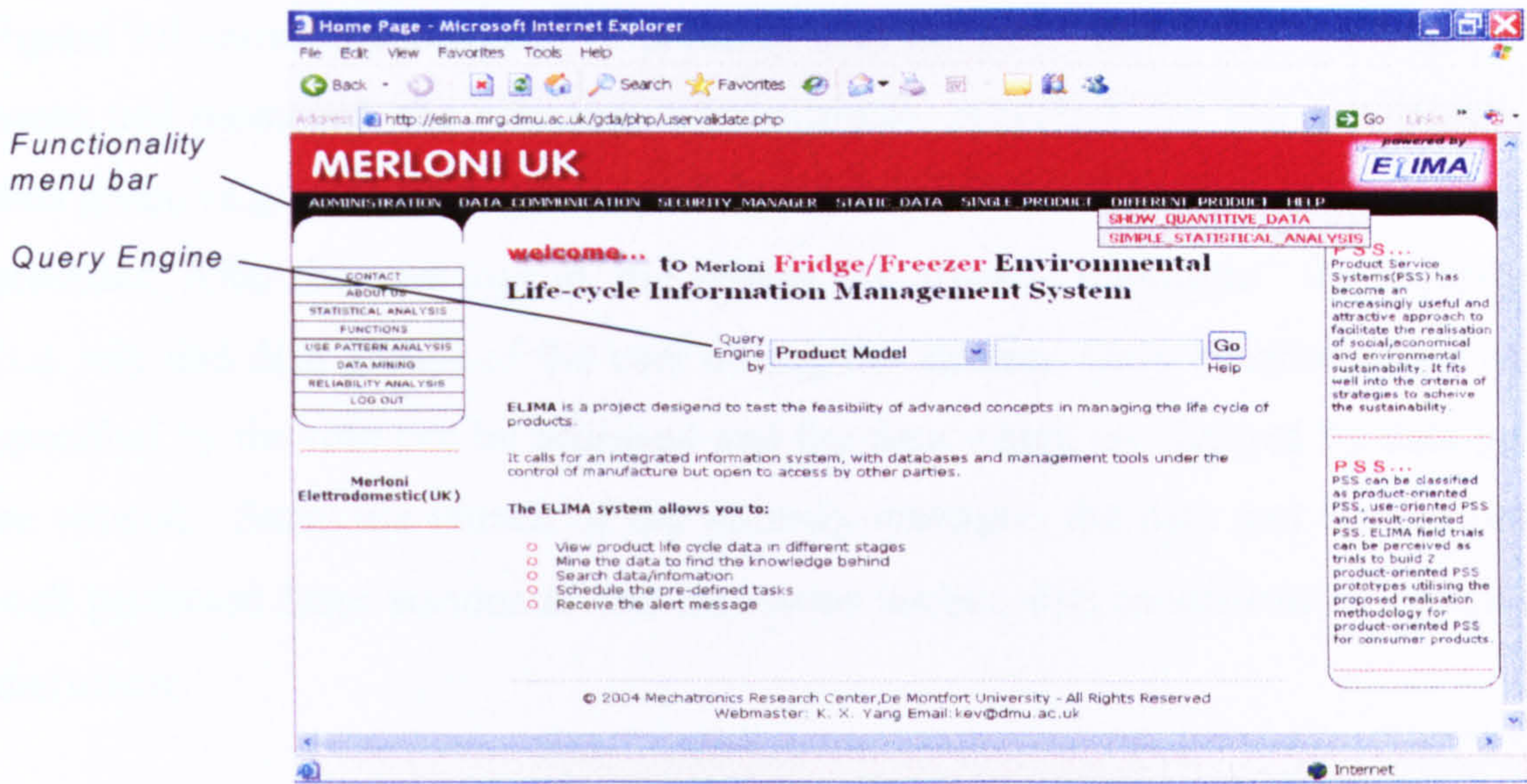


Figure 9.8 A screenshot of the FFIMS interface

9.3.3.2 Security Management

FFIMS houses sensitive product life cycle data. The *security manager* is responsible for the security of the system and prevents it from accidental and malicious access, use, modification, destruction, or disclosure. The *security manager* created from the Framework fully meets the requirements specification. Apart from the normal user name-password authentication, it supports 3-level security control (i.e. user access control, functionality access control and data access control) and role-session paradigm (i.e. same user with different role get a different view of the system).

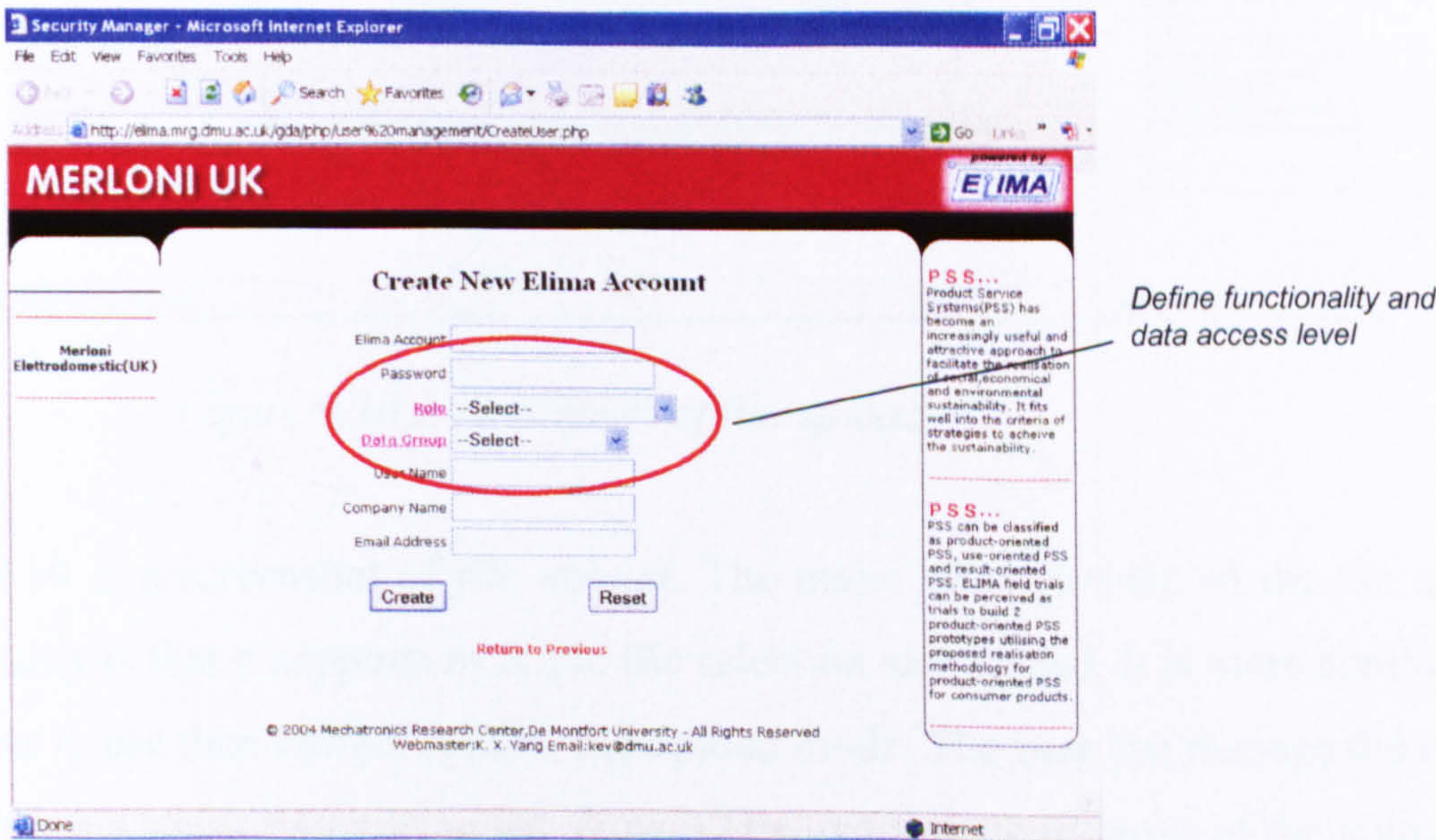


Figure 9.9 A screenshot of creating a user account



Figure 9.9 shows a screenshot of creating user account. Apart from providing account name and password, the role (e.g. administrator, recycler) that the user plays and the data group (e.g. all dynamic data, part dynamic data) that the user can access have to be provided. After the user logs in, the system can always “memorise” the security profile (i.e. role and data group) of the user during the session. Only functionalities which are specified by the role can be accessed and the data which are defined by data group can be viewed. Since the launch of the security manager, the data and system have been well protected from accidental and malicious access, use, modification, destruction, or disclosure.

9.3.3.3 Data Communication: File Upload

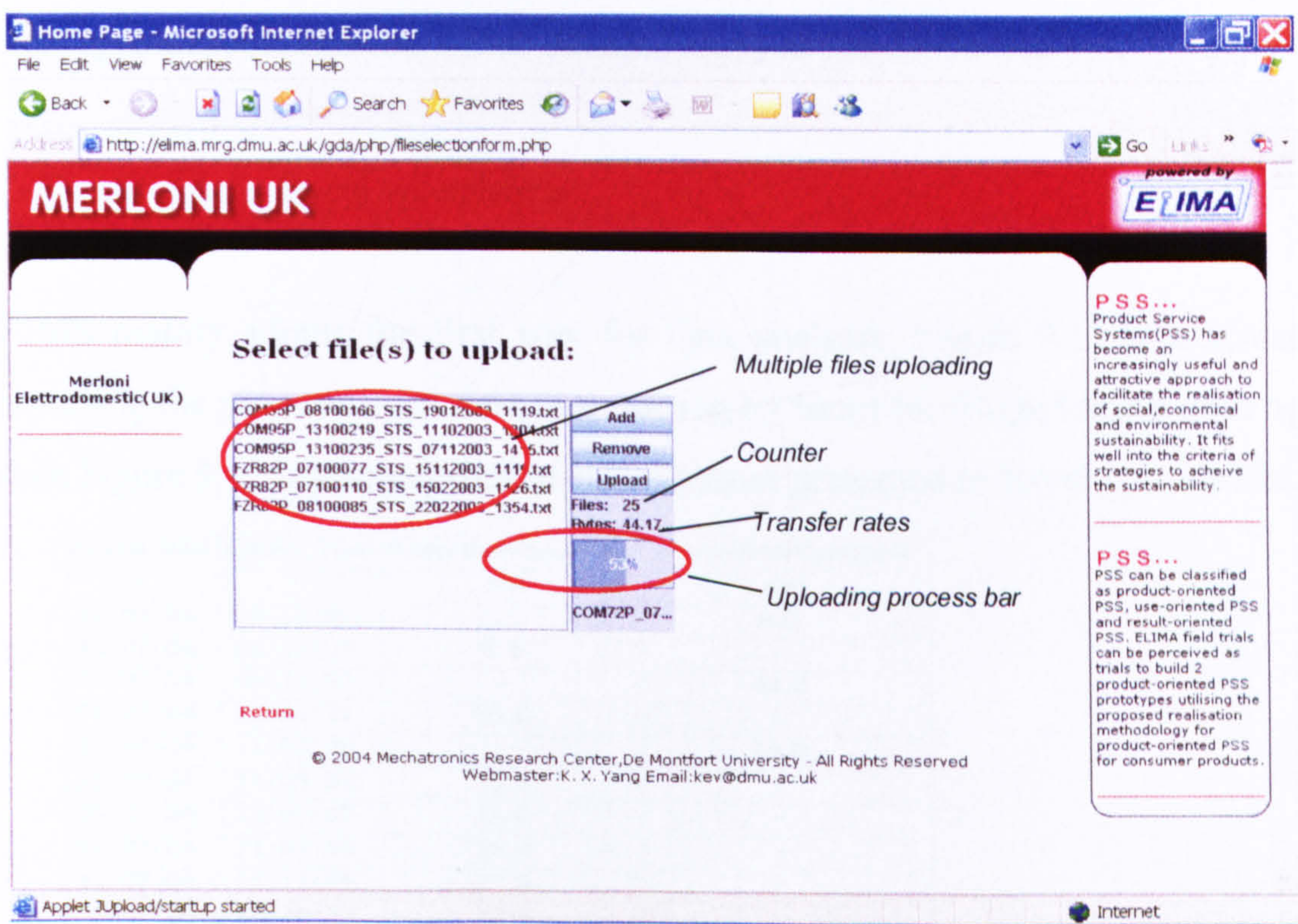


Figure 9.10 A screenshot of file upload

Figure 9.10 is a screenshot of *file upload*. The major characteristic of the *file upload* functionality is that it supports multiple file selection and upload. It is more comfortable and easier to use than normal HTML file upload mode. The user can manage the upload queue. It has a status monitoring bar showing the percentage increase of the upload and has the current file transfer rate in kilobytes per second. In white goods field trial, 144



files from 89 fridge/freezers have been successfully uploaded to the FFIMS through using *file upload*, while in brown goods field trial 301 XML readouts from 205 PS2 units have been successfully uploaded to the PSIMS through using *file upload*.

#### 9.3.3.4 Data Analysis

The Framework offered two major ways for data analysis, (i) pre-processing of data and export of data to Excel format for further analysis, (ii) online data analysis using the built-in statistical functions, which include commonly statistical functions such as standard division, average, correlation, histogram, etc. For simple analysis, the online data analysis tools have been demonstrated to be useful for ELIMA partners. However, for detailed analysis, the partners found that it was more useful to export the data to Excel spreadsheet based on which detailed analysis can be carried out.

- Export data to Excel spreadsheet

FFIMS mainly adopts the first way for data analysis. Figure 9.11 is a screenshot of displaying the pre-processed data (e.g. sorting by time) for fridge/freezer door open time; while Figure 9.12 is a screenshot of same dataset presented in Excel spreadsheet.

Collect_Date	Collect_Time	FridgeDoorOpenTime	FreezerDoorOpenTime
-	-	sec	sec
31-07-04	09:12:40	-	9.0
31-07-04	09:15:27	4.0	-
31-07-04	09:39:57	-	48.0
31-07-04	11:01:21	95.0	-
31-07-04	11:02:30	-	16.0
31-07-04	11:03:52	25.0	-
31-07-04	11:07:07	11.0	-
31-07-04	11:07:58	25.0	-
31-07-04	11:13:46	27.0	-
31-07-04	11:36:53	15.0	-
31-07-04	13:19:58	4.0	-
31-07-04	13:23:31	96.0	-
31-07-04	13:25:52	59.0	-
31-07-04	13:27:49	67.0	-
31-07-04	15:05:33	-	24.0
31-07-04	15:25:45	-	6.0
31-07-04	16:03:10	4.0	-
31-07-04	16:03:22	3.0	-
31-07-04	16:08:51	-	14.0
31-07-04	16:24:57	32.0	-
31-07-04	16:28:45	90.0	-
31-07-04	16:32:57	10.0	-
31-07-04	16:34:21	9.0	-

Figure 9.11 Snapshot of dynamic data display in HTML table



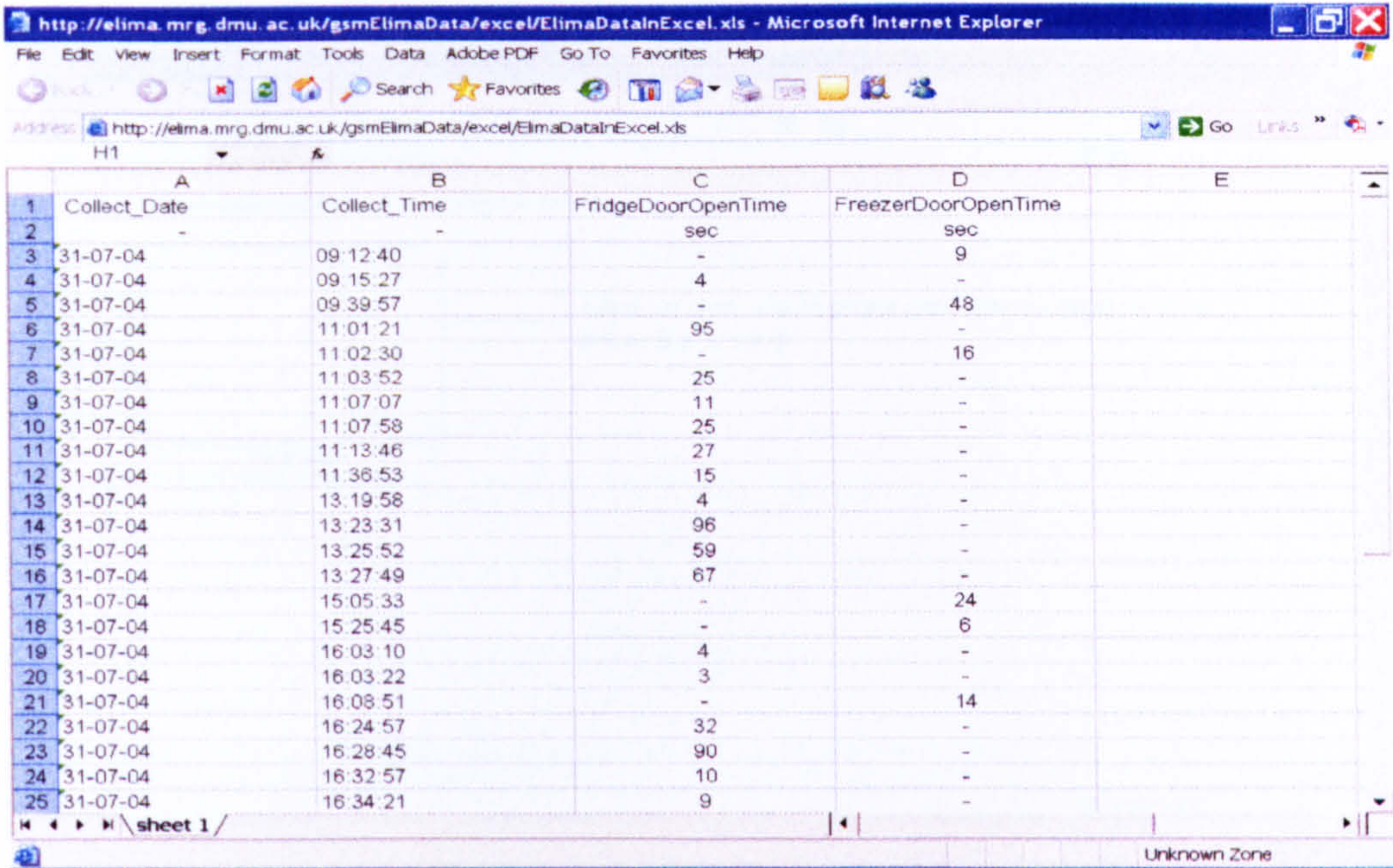


Figure 9.12 The export of dynamic data to Excel spreadsheet

- On-line Data Analysis

On-line data analysis is implemented in FFIMS. Figure 9.13 and Figure 9.14 are two screenshots of on-line data analysis results presented in charts.

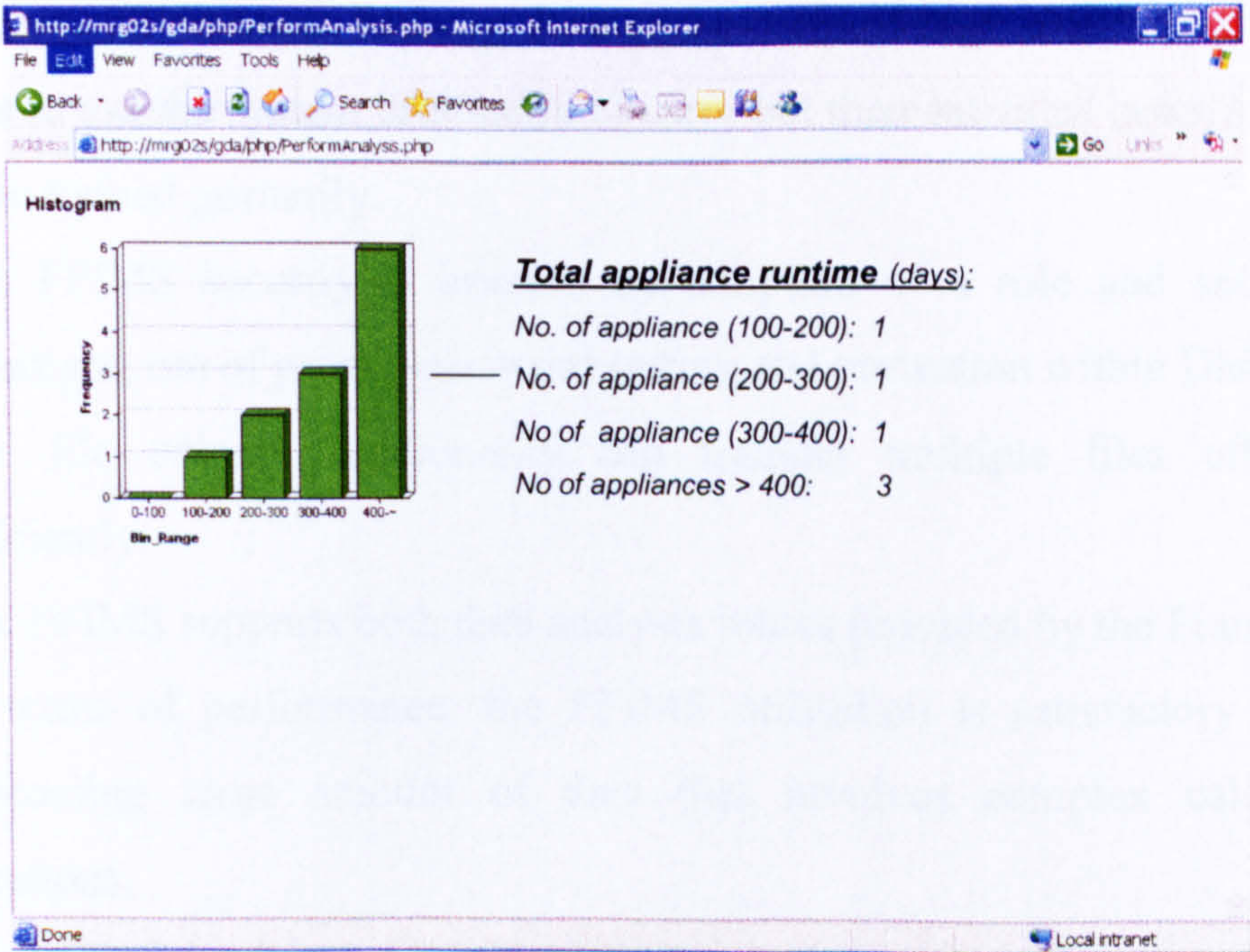


Figure 9.13 A screenshot of on-line analysis: histogram



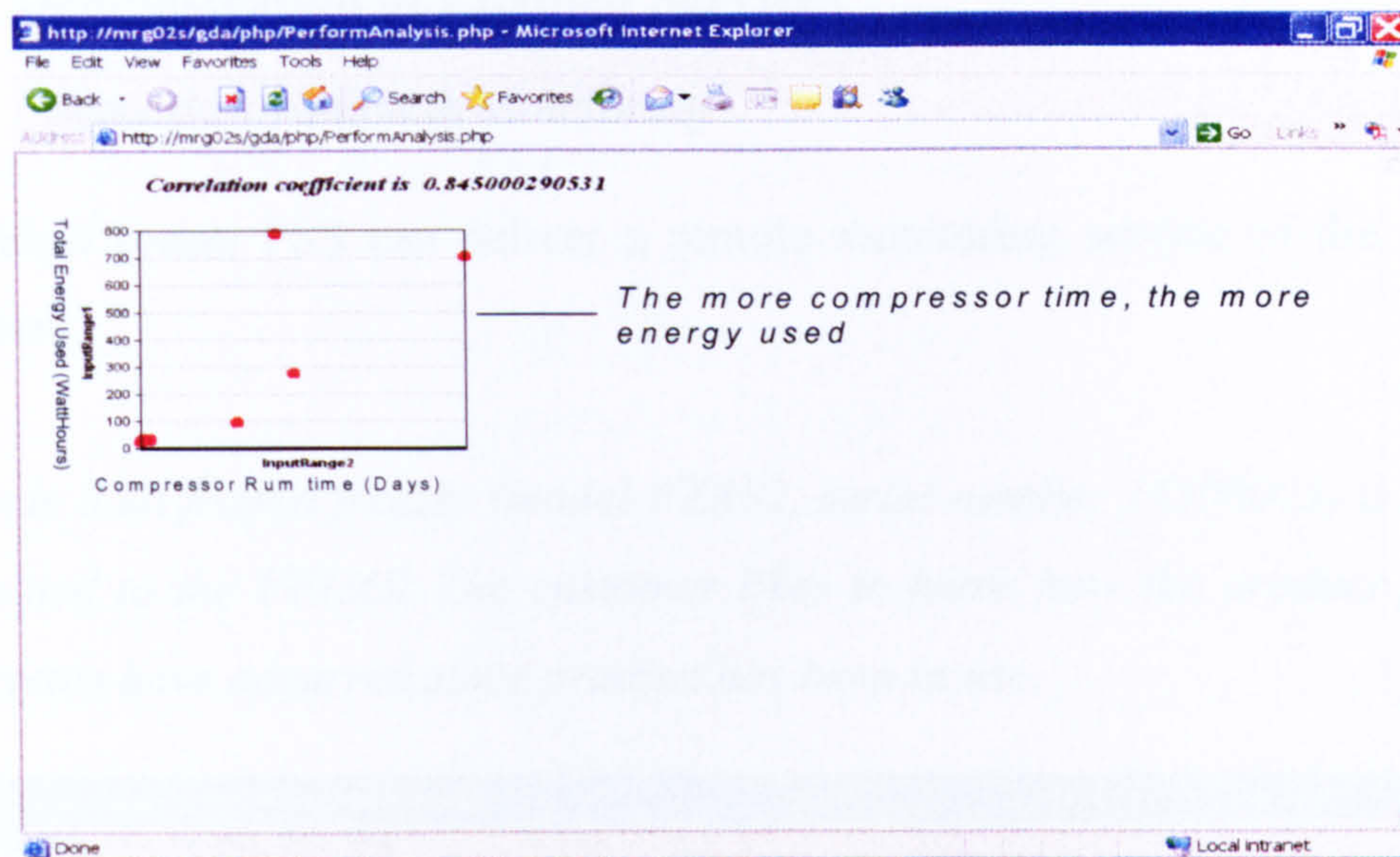


Figure 9.14 A screenshot of on-line analysis result: correlation

#### 9.3.3.5 Evaluation Result

The evaluation result of FFIMS developed by using the Framework is summarised as follows:

- 1) The user interface is friendly. FFIMS has offered high usability where users are able to use the system efficiently to carry out their intended tasks without having to be trained primarily.
- 2) The FFIMS security is assured by adoption of a role and session security paradigm, use of proper password setting and protection within DMU firewall.
- 3) The file upload functionality can transfer multiple files effectively and efficiently.
- 4) The FFIMS supports both data analysis routes provided by the Framework.
- 5) In terms of performance, the FFIMS utilisation is satisfactory except when processing large amount of data that involves complex calculations and iterations.
- 6) The FFIMS is also highly reliable with almost no downtime reported during the later stage of the system except when carrying out the planned maintenance and upgrade.



9.3.4 Demonstration of Enabled Services

9.3.4.1 Scenario 1 - Remote Monitoring

The fridge/freezer PSS can deliver a remote monitoring service of the appliance for customers.

*Life cycle data from a freezer (model FZR82, serial number 14100075) is acquired and transmitted to the FFIMS. The customer likes to know how the product performs and what events have occurred since product has been in use.*

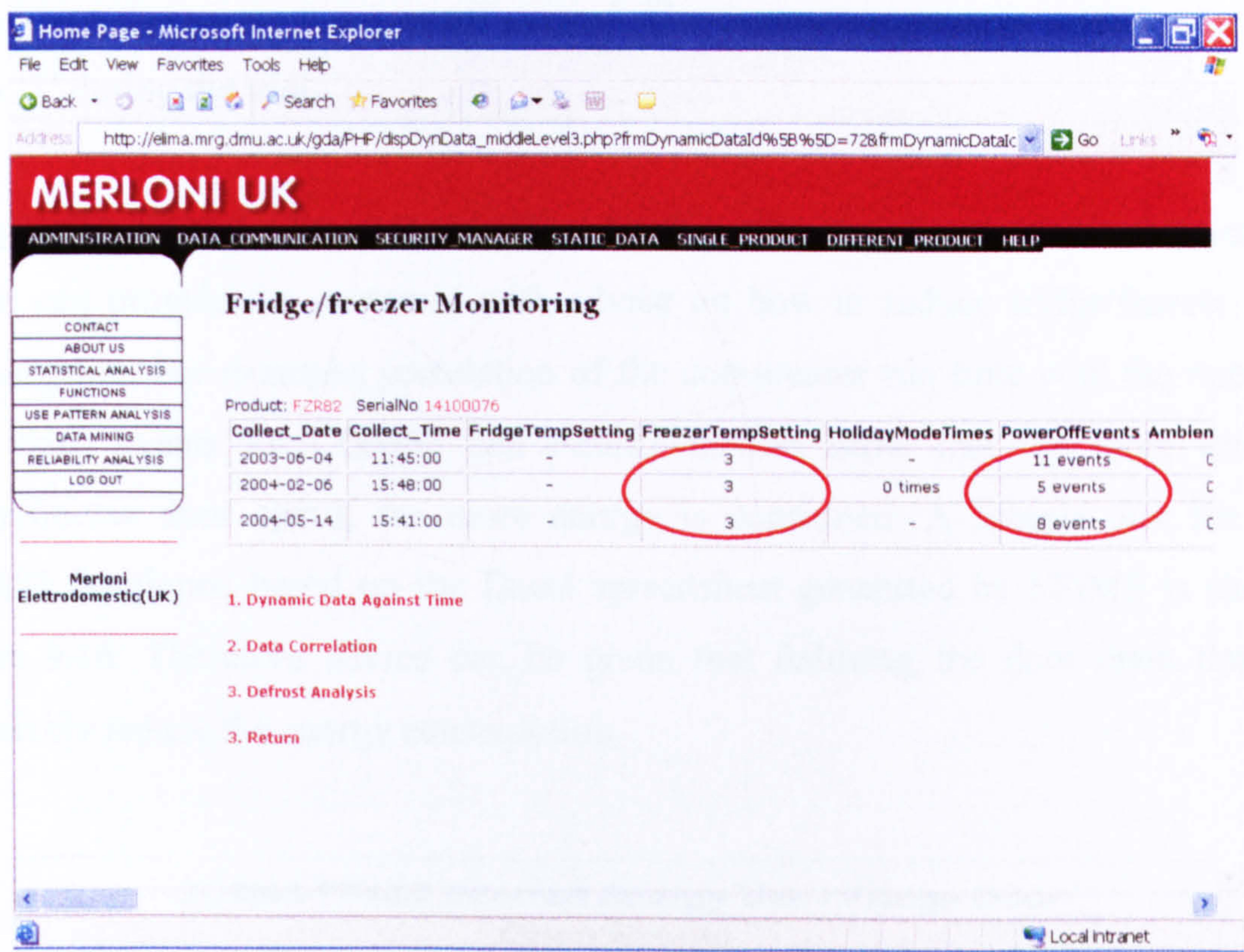


Figure 9.15 A screenshot of freezer monitoring information

Background: In this field trial, 89 participants at level 2 and 6 participants at level 3 agreed to have a log taken of the life cycle data of their fridge/freezers. Apart from logging fridge/freezer specific data (i.e. compressor run time, door open times), the fridge/freezer PSS also logged energy consumption data and temperature values every minute, and logged current and voltage values every 3 hours. All the data gathered was transmitted to FFIMS. Analysis of these data allows the profile of the fridge/freezer usage to be established. The *security manager* of FFIMS ensures the customers can



access to information concerning their fridge/freezers (i.e. energy consumption, surrounding environment) and events that have arisen (i.e. sensor faults, power off) at any time and from any place where they can access the Internet.

Remote monitoring: From the life cycle data acquired and its transmission to FFIMS, the freezer (model FZR82, serial number 14100075) is monitored throughout the field trial. Figure 9.15 is a screenshot of the freezer remote monitoring information. It shows that from the field trial start date to 2003-06-04, there were 11 power off events; from 2003-06-04 to 2004-02-06, there were 5 power off events and from 2004-02-06 to 2004-05-14 there were 8 power off events<sup>7</sup>. The freezer temperature setting was always set to ‘3’ during the trial.

Energy consumption advice: One of the major benefits of a remote monitoring service is that it can provide the customer with advice on how to reduce fridge/freezer energy consumption. For example, correlation of the compressor run time with the number of door open events for COM95 and FZR82 models show that the more times the fridge/freezer door opens, the more energy is consumed. A sample plot for model COM95 developed based on the Excel spreadsheet generated by FFIMS is shown in Figure 9.16. Therefore advice can be given that reducing the door open times can effectively reduce the energy consumption.

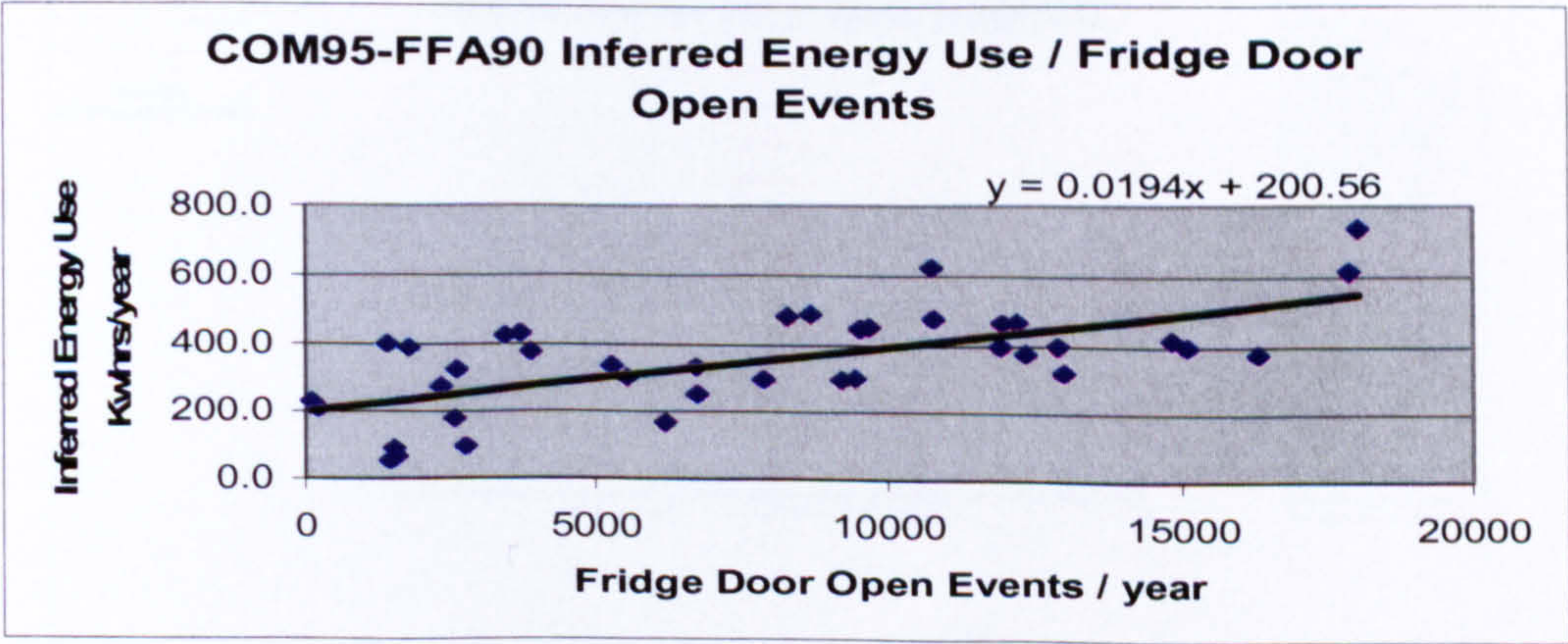


Figure 9.16 Compressor cycles v fridge door openings (ELIMA Seminar, 2005)

<sup>7</sup> Some trial fridge/freezers were often powered off during the field trial



9.3.4.2 Scenario 2 - End-of-Life Treatment

The fridge/freezer PSS can enable recycler/refurbisher and manufacture to provide better fridge/freezer end-of-life treatment service to comply with WEEE and RoHS Directives.

*In order to dispose of the end-of-life fridge/freezer, a recycler or refurbisher likes to know the information of internal components within a fridge/freezer.*

Background: With the availability of BoM, information about hazardous substance can be identified and made accessible to all the stakeholders involved. With the BoM and dynamic life cycle data, the end-of-life product/component remaining life time can be predicted, and parts can be removed and classified for treatment according to their life histories and material content, etc.

As the fridge/freezer FFIMS houses BoM for 8 product models, the authorised users can easily access BoMs through the FFIMS to get the product information (Figure 9.17).

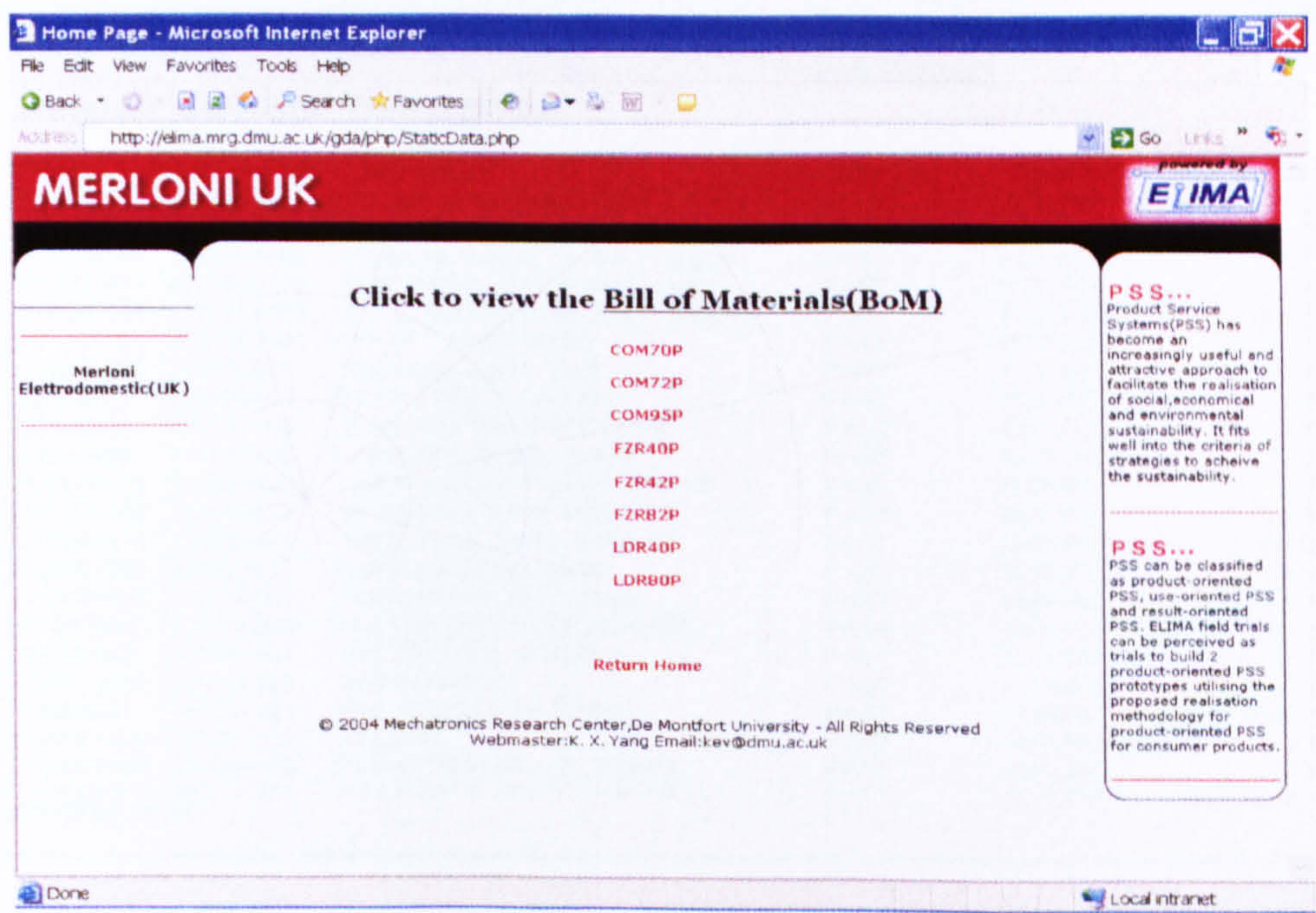


Figure 9.17 Access BoM through FFIMS



End-of-life treatment: The BoM of the COM95 can help recycler/manufacture to dispose the end-of-life COM95 fridge/freezer. A screenshot of BoM for COM95 displayed by FFIMS is shown in Figure 9.18. Through the BoM, the following information can be acquired to support the end-of-life COM95 fridge/freezer treatment:

- A. The information of internal components within COM95 fridge/freezer can be acquired. For example, FA90P FOAM CAB ASSY (code, 2731001501), SIDE PANNELWHITE L/H (1989) (code: 2710200008) etc.
- B. The relationship between internal components. For example, the component 2731001501 is the parent component of 2710200008.
- C. Fridge/freezer refurbishment is another way of end-of-life treatment. As BoM also listed parts name, this can guide refurbishers to purchase parts online at the relevant website (e.g. HotPoint Service website).

Part Number COM95P Description								
Level	Parent	Component	Description	Effect In	Effect Out	Qty Per	Class	Iti
1	COM95P	2731001501	FA90P FOAM CAB ASSY	PAST	FUTURE	1	F	
2	2731001501	2710200008	SIDE PANEL WHITE L/H (1989)	PAST	FUTURE	1	F	
3	2710200008	1120001068	STEEL BLN WHT 3914 X 719 ELK	PAST	FUTURE	0.5	P	
2	2731001501	2710200108	SIDE PANEL WHITE R/H (1989)	PAST	FUTURE	1	F	
3	2710200108	1120001068	STEEL BLN WHT 3914 X 719 ELK	PAST	FUTURE	0.5	P	
2	2731001501	2730001601	FFA90'S LINER ASSY	PAST	FUTURE	1	F	
3	2730001601	229106801	TRANSOM ASSY WHT	PAST	FUTURE	1	F	
4	229106801	229204403	FOAM TAPE 560X7MM	PAST	FUTURE	2	P	
4	229106801	229207103	TRANSOM BRACKET WHITE	PAST	FUTURE	1	F	
5	229207103	111940480	LAMINATE STEEL BLANK	PAST	FUTURE	1	P	
3	2730001601	2730001600	LINER ASSY (FFA90) C/PARTS	PAST	FUTURE	1	F	
4	2730001600	124940474	38MM TAPE PP64 4089	PAST	FUTURE	0.4	P	
4	2730001600	126982475	19MM TAPE (BREATHABLE)	PAST	FUTURE	0.8	P	
4	2730001600	126983777	36MM MASKING TAPE	PAST	FUTURE	1.8	P	
4	2730001600	225256901	FOAM STRIP 100 X 25MM	PAST	FUTURE	1.5	P	
4	2730001600	228264002	GUTTER DRAIN (F/F)(EIGER)	PAST	FUTURE	1	P	
5	228264002	045557991	POLYSTYROL NATURAL	PAST	FUTURE	0.0113	P	
4	2730001600	228269001	DRAIN SPOUT	PAST	FUTURE	1	P	
5	228269001	045557991	POLYSTYROL NATURAL	PAST	FUTURE	0.0093	P	
4	2730001600	229203001	COLLAR	PAST	FUTURE	2	P	
4	2730001600	229204406	FOAM TAPE 50 X 20 X 6MM	PAST	FUTURE	2	P	
4	2730001600	229204412	FOAM TAPE 95 X 40 X 6MM	PAST	FUTURE	1	P	

Figure 9.18 A screenshot of BoM for COM 95



### 9.3.5 Summary of Test Case I

- 1) The fridge/freezer PSS prototype has been developed by utilising the proposed PSS realisation methodology and associated toolkit.
- 2) An instance of the service enabler (i.e. FFIMS) created by employing the Framework has been evaluated by integrated into the ELIMA white goods field trials. The evaluation shows that the FFIMS developed by using the Framework can meet the user requirements in terms of interface, security, data transfer, data analysis and service enabling.
- 3) The following services have been delivered via the fridge/freezer PSS prototype.
  - Remote monitoring service
  - Product-end-life treatment service

## 9.4 Test Case II: PlayStation 2 PSS Prototype

As functionalities of PSIMS created by using the Framework have many similarities with FFIMS, the evaluation of PSIMS will not be discussed. Hence in this test case, we will:

- Use the Framework to create PSIMS and integrate it into the trial.
- Demonstrate services which are delivered via the PSS prototype.

### 9.4.1 ELIMA Brown Goods Field Trial Overview

ELIMA brown goods field trial is undertaken by *Sony International (Europe) GmbH, Stuttgart, Germany*, which is a member of ELIMA consortium.

#### 9.4.1.1 Trial Product and Specification

The game console PS2 is used as brown goods exemplar product. PS2 (Figure 9.19) is a computer entertainment system which incorporates both CD and DVD formats, and



provides games, video and music. It mainly consists of a console and an analogue controller, which is connected to the console via a cable. Picture is displayed through a normal TV set. The console can be operated either in horizontal or vertical position and is usually used under normal living room conditions. The PS2 has a main power switch but can also left in a stand-by function. For this trial, a standard PS2 has been modified with addition of an IDU (ELIMA D25, 2004).



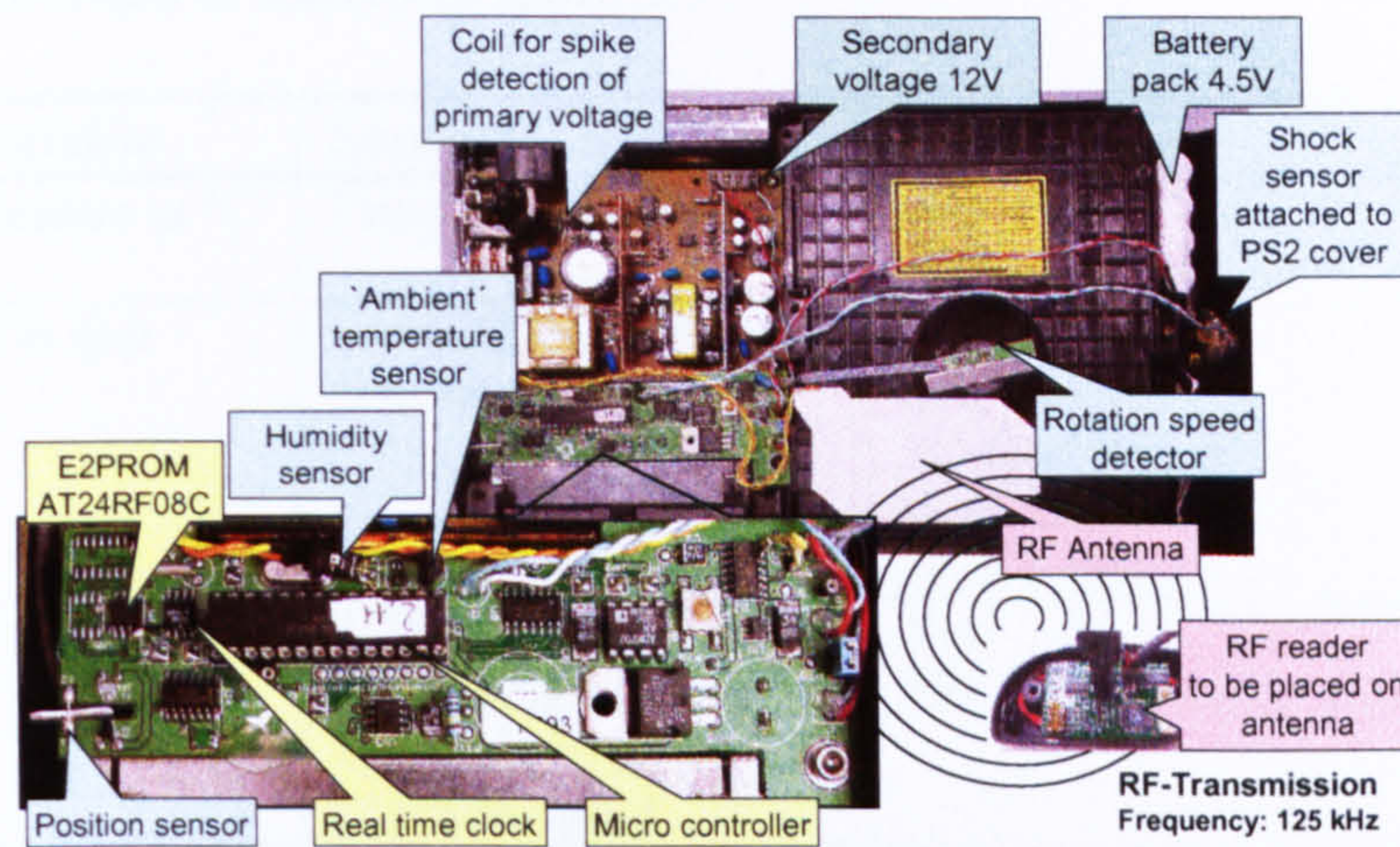
*Figure 9.19 Trial product: PlayStation2*

In order to obtain realistic data and as close to real situation as possible, the trial were arranged among about 200 Sony employees, separated into groups of about 50 users in four countries (Germany, UK, Spain and Italy). The field trial duration is between 5 and 11 months. At the end of the field trial, customers are required to replace the ELIMA PS2 with a normal PS2 at the service centre and they will get a PS2 game for free. The collected data will then be retrieved and analysed.

#### 9.4.1.2 IDU and Data Acquisition

Figure 9.20 give an overview of IDU embedded into PS2 game console to acquire the life cycle data (ELIMA D25, 2004). Recording of dynamic life cycle data requires sensors or other recording functions. Data recording occurs when PS2 is switched on or in the stand-by mode (secondary 12V from the consoles power supply available). Shocks during transportation can be recorded even when the PS2 is switched off where the power is supplied by an attached battery.





*Figure 9.20 PlayStation2 IDU specification (ELIMA D25, 2004)*

Implemented sensors are:

- Humidity sensor
- 2 temperature sensors. One is used to measure the ambient temperature in the console and the other is used to measure the temperature in the heat sink.
- Position sensor (horizontal / vertical position)
- CD/DVD speed detection
- Shock sensor
- Spike detector
- Secondary voltage control



Data to be recorded is shown in table 9.1

Recorded variable	Sensor(s) applied	Counter	Event based
Ambient temperature in °C	Temperature sensor	No	Date:time
Rotation speed in rpm	Optical CD drive speed detection	Yes: Low Speed Medium Speed High Speed all in seconds	Date:time & max. speed recorded in an operation cycle
PS2 On	2 Temperature sensors (ambient & heat sink temperature) or CD drive speed	Yes: PS2 On time in seconds	Date:time
Spikes on secondary voltage	Coil for spike detection	Yes: Number of spikes	Date:time
Operation in horizontal or vertical position	Optical detection of operation mode	Yes Vertical & horizontal operation in seconds	Date:time of each change
Secondary voltage in V	Internal voltmeter	Yes: Operation in seconds	Date:time
Humidity in %rH	Humidity sensor	No	Date:time
Mechanical shocks	Shock sensors	Yes: Number of shocks 3 levels (1,2,3)	Date:time

Table 9.1 Life cycle data to be recorded

Table 9.2 shows the data of standard event to be recorded

Parameter	How obtained	Flag condition/reason
No. of standard events	From standard event list	
Standard event drop outs	Lines with no event/no date/no value in the body of the std event list	>0
No. of PS2 On events	From standard event list	>30 Indicates a speed detection problem
No. of horizontal events	From standard event list	
No. of vertical events	From standard event list	
No. of low speed events	From standard event list	>30 Indicates a speed detection problem
No. of medium speed events	From standard event list	
No. of high speed events	From standard event list	

Table 9.2 Data of standard event to be recorded



Table 9.3 shows the data of dramatic event to be recorded

Parameter	How obtained	Flag condition
<sup>8</sup> No. of shock level 1	From shock level 1 records	No
Newest shock level 1	Date shock level 1 records	n.a.
<sup>9</sup> No. of shock level 2	From shock level 2 records	No
Newest shock level 2	Date shock level 2 records	n.a.
<sup>10</sup> No. of shock level 3	From shock level 3 records	>0
Newest shock level 3	Date shock level 3 records	n.a.
No. of over voltage events	From voltage records	>0
Newest voltage event	Date & value from voltage records	
Most severe voltage event	Date & value from voltage records	> 13V
No. of humidity events	From humidity records	>0
Newest humidity event	Date & value from humidity records	
Most severe humidity event	Date & value from humidity records	> 90%rH
No. of temperature events	From temperature records	
Newest temperature event	Date & value from temperature records	
Most severe temperature event	Date & value from temperature records	> 80°C

Table 9.3 Data of dramatic event to be recorded

9.4.1.3 Data Flow

The overall information flow of brown goods field trial is shown in Figure 9.21. The left side schematically illustrates the information flow of the dynamic data from the IDU to the PSIMS. On the right side the flow for the static data is displayed. Some trial specific information (e.g. trial start date, trial location) is collected through the product registration (ELIMA Seminar, 2005).

For the static data, two treatment documents are prepared to put in PSIMS. One document is *Information for Treatment Facilities*, which describes the general disassembly procedures and the location of substances that have to be removed according to WEEE. This document is intended to be available to all recyclers. The other is *Extended Information for Treatment Facilities* which contains more details about disassembly, potential value of components, reuse information, proposed best treatment technologies as well as the information required by WEEE.

<sup>8</sup> Shock level 1 means hard placement  
<sup>9</sup> Shock level 2 means the kick-over from vertical use position to horizontal position  
<sup>10</sup> Shock level 3 means waist height drop



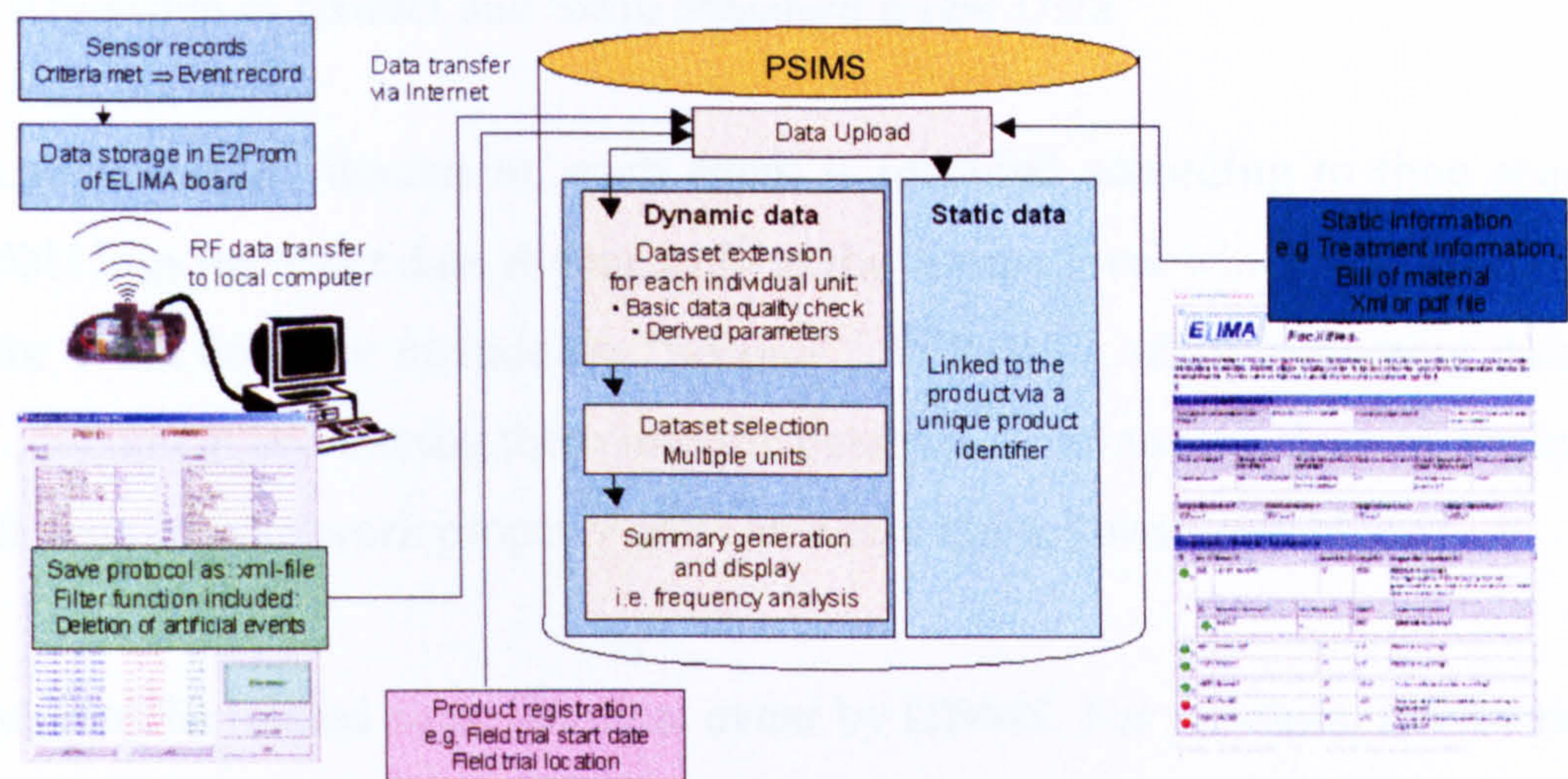


Figure 9.21 Brown goods field trial data flow (ELIMA Seminar, 2005)

The mixture of event based sensor recording and counters are chosen for dynamic data recording. As soon as a sensor signal meets the set criteria (e.g. exceeds the set limit) it is recorded in the standard event (i.e. PS2 on) section or dramatic event (i.e. shock event) section and stored on the EEPROM. RFID technology is used to read the data from EEPROM to a local computer. *ATMEL AT24RF08-EK Quick Start Kit*<sup>11</sup> and the appropriate software are used to read the data out from the EEPROM and saved into a local computer as an XML document. The RF reader reading distance is about 1cm and reading 8k bytes takes about 15 seconds. By using the RFID, the data can be retrieved even PS2 is defect. The XML document is then transmitted to the PSIMS by using file upload facility provided by PSIMS.

#### 9.4.2 Using the Framework to create PSIMS

The PSIMS skeleton is created by using the Framework. In order to support implementations of PS2-specific functionalities, some algorithms have been developed.

<sup>11</sup> A software package for AT24RF08C, which is a dual access EEPROM with both a wired serial port and a wireless RFID port used to access the memory.



### 9.4.2.1 Algorithm to Extract and Store Standard Event Data

In the generated XML document, each event is recorded according to time sequence. But the IDU logs the event date & time only at the minute level which results in the date & time for event does not include the “second”. At PSIMS, when extracting data from the XML document and storing them into the database, common routines of sorting data by date & time can not work properly and can cause the following problems:

- 1) Events may be treated as an identical event by DBMS. For example, two events (e.g. medium speed), one happened at 2004-03-05 12:34:34 and the other happened at 2004-03-05 12:34:55 are both recorded by IDU as events happened at 2004-03-05 12:34:00. When these two data is stored into the database, the DBMS may treat them as identical events accepting one and discarding another.
- 2) Standard events happen within a single minute (same date & time) may not be displayed correctly. But displaying standard event in this order is particularly important for use pattern analysis.
- 3) As the standard event is logged in a circular way (i.e.  $\text{DateTime1} = \text{DateTime64}$ ) at IDU, events which have the same date & time from the top and bottom of the standard event list can cause event data display in a wrong order.

This often happens to standard event. In order to tackle problems caused by inadequate resolution of the date & time, we proposed an algorithm to extract data and store them into database in a time sequence order (Figure 9.22). Assume there are 64 standard events. Considering the feature that the standard event is logged according to time sequence and in a circular way, the proposed algorithm takes the first entry  $E_1$  from the XML document as a reference, search the earliest event  $E_i$  by comparing each subsequent event with  $E_1$  and save the detected  $E_i$  to the database followed by events from  $i$  to 64. If the latest event  $E_{i-1}$  is found, save the  $E_{i-1}$  to the database followed by events from 1 to  $i-1$ . Detail description of the algorithm is illustrated as follows:



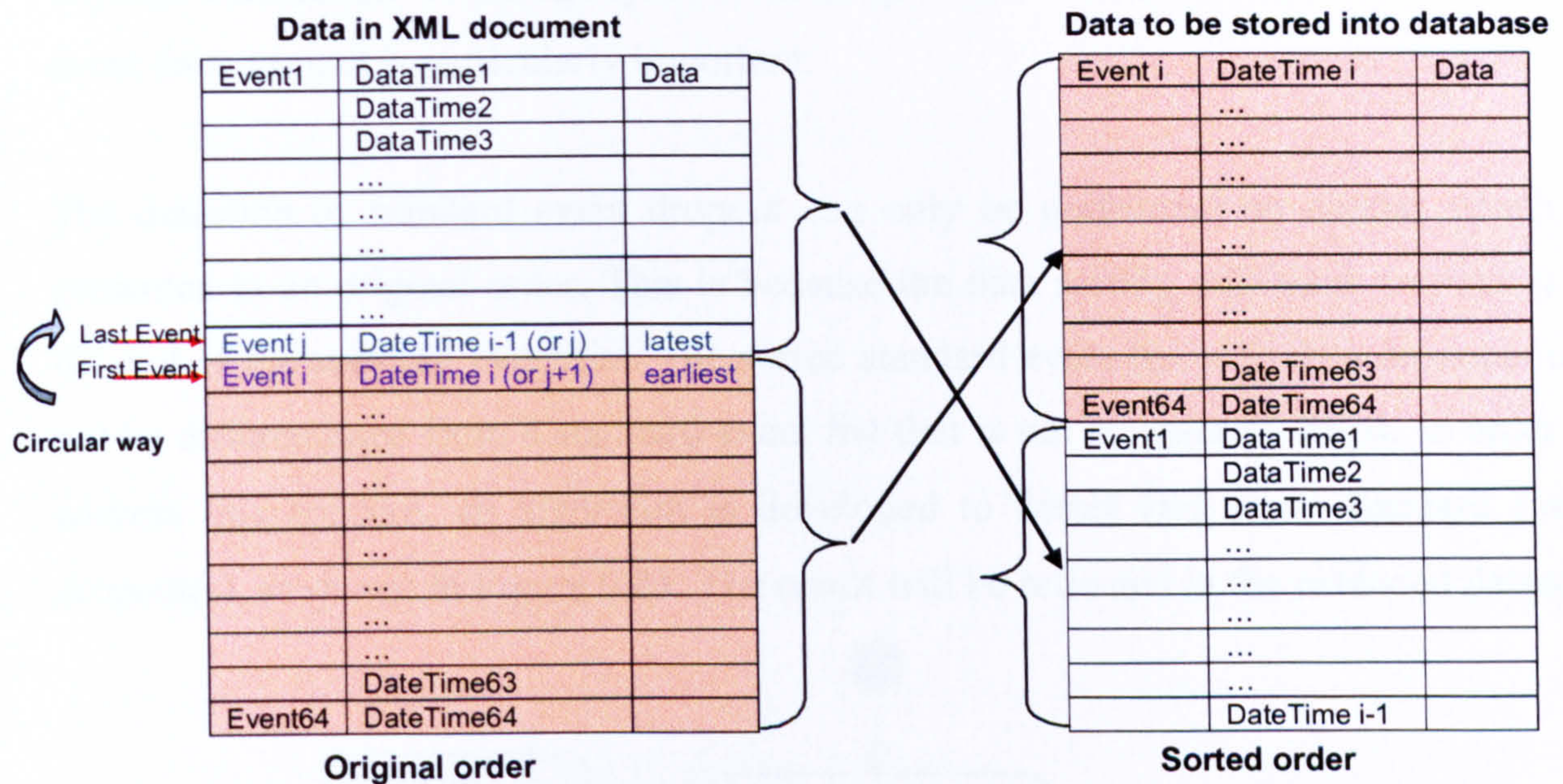


Figure 9.22 Data extraction algorithm

- Take DateTime1 as reference
- Search for the earliest time (time i) from the beginning of the list
- If DateTime i < DateTime1, then save events to database, starting with events i to 64, follows by events 1 to i-1
- If DateTime1 is the earliest, then events 1 to 64 will be written to the database, discarding the null events
  - Else search for the latest time (time j) from the end of the list
  - If DateTime j > DateTime1, then save events to database, starting with events j + 1 to 64, follows by events 1 to j
  - If DateTime64 is the latest, then events 1 to 64 will be written to the database

#### 9.4.2.2 Algorithm to Determine the Standard Event Dropout(s)

Data dropout is an incomplete standard event which has no date, no event type and no value recorded in the XML document. We can not treat this incomplete event as nothing happened as it does have special meaning. For example, as this “no date, no event type and no value ” event also occupies one storage space, knowing data dropout can help us



to judge whether the 64 storage space is filled up or not. Hence the detection of standard event data dropout is particularly important.

The detection of standard event dropout can only be performed on dataset which is presented in an original order. This is because the data sorting will send dropout(s) to the end of the standard event list. The sorted standard event list with data dropouts can not be differentiated from a standard event list that is not completely filled. In order to address this concern, an algorithm is developed to detect and count standard event dropout(s), as shown in Figure 9.23. The result will be reflected in the extended dataset.

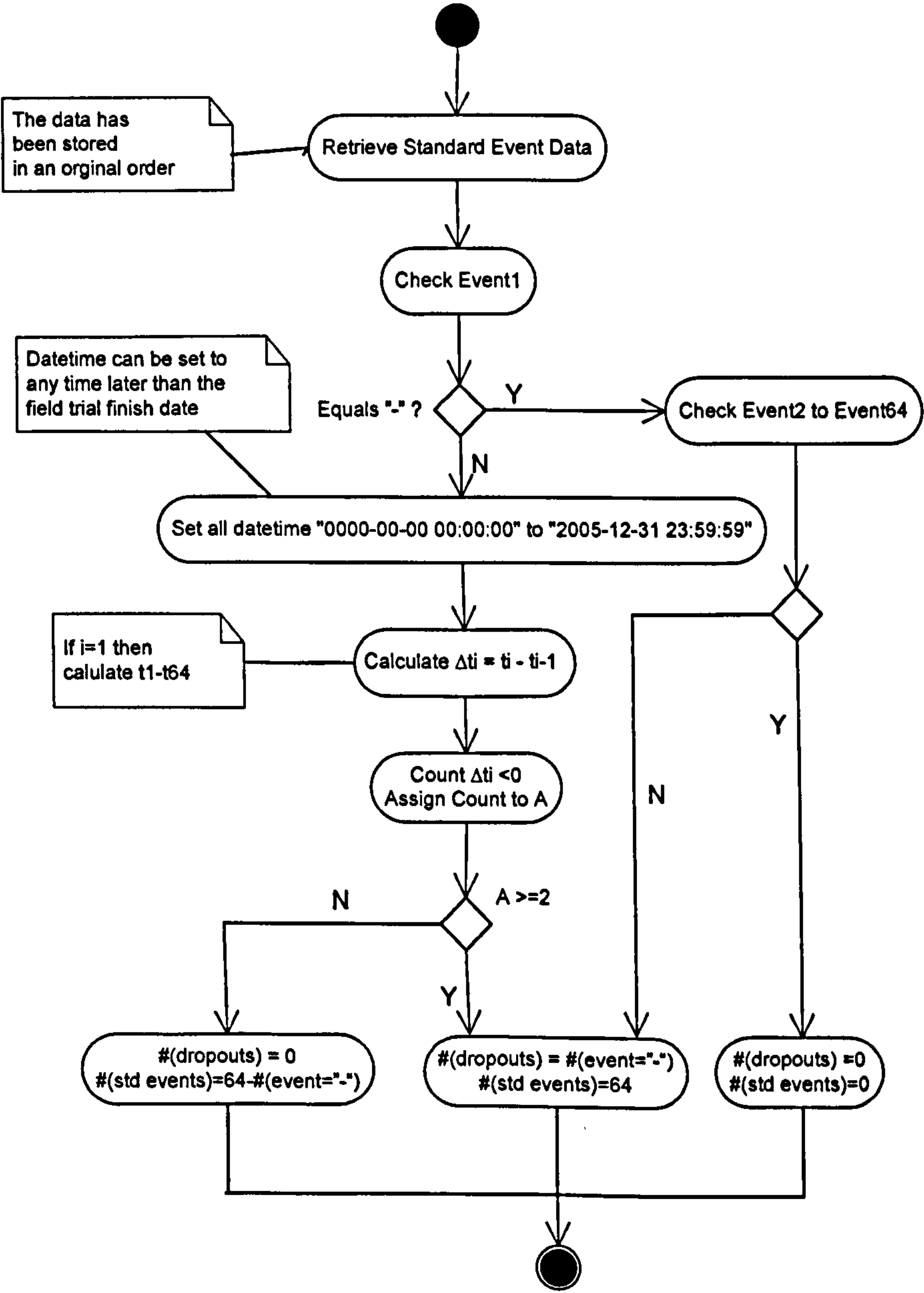


Figure 9.23 Algorithm to determine the number of standard event dropout(s)



The proposed algorithm can also be used for the following calculation related to the standard event:

- Original position of the newest standard event
- Number of standard events
- Maximum time between two standard events

#### 9.4.2.3 Algorithm to Parse the XML Document

The data readout from the PS2 is presented as an XML document. When this XML document is uploaded to the PSIMS, an XML parser is required to parse the XML document. However, this parsing task is not just a normal XML document parsing as it involves the following operations:

- 1) Extract the data from XML document
- 2) Store the extracted data in a sorted order and original order separately.

As the calculation of the extended data of *standard events summary* requires the data to be stored in an original order so that *standard event dropouts*, *original position of the newest standard event*, *number of standard events* and *maximum time between two standard events* can be derived. In the meanwhile, calculations of other extended data (i.e. *field trial basic data*, *quality check* and *severe event summary*) require data to be stored in a sorted order by using the data extraction and storing algorithm proposed previously.

- 3) Calculate the extended dataset

In order to address this need, an XML parsing algorithm is proposed to parse the XML document of PS2 data readout, as shown in Figure 9.24.



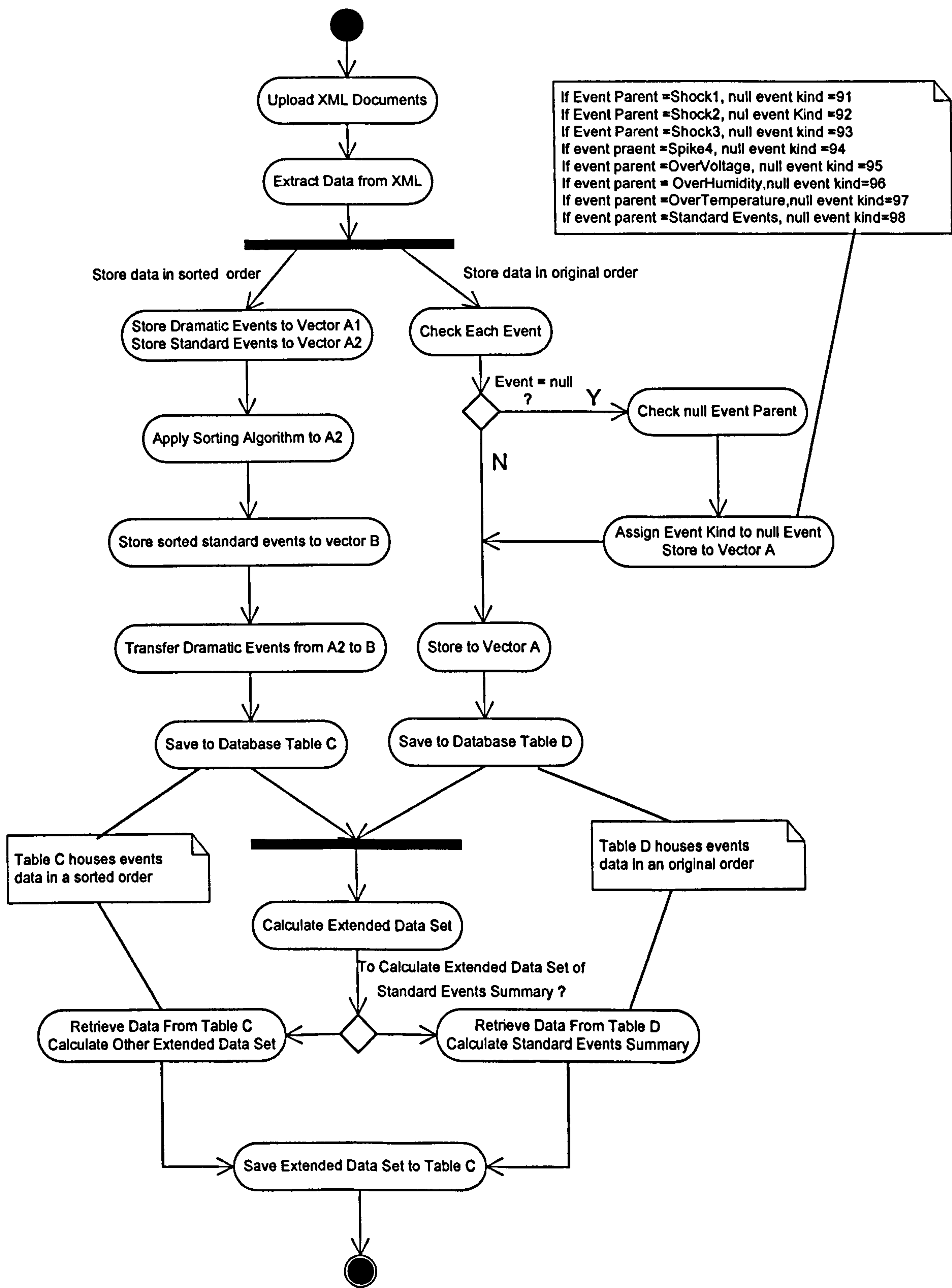


Figure 9.24 Algorithm of an XML parser for PS2 data readout



9.4.3 Demonstration of Enabled Services

9.4.3.1 Scenario 1 - Remote Diagnostics

The PS2 PSS prototype can deliver a remote diagnostics service.

*A user reports that his PS2 experienced a disturbance (PS2 serial No: 0719952J136). But it is unknown whether this shock has damaged the CD-drive. Assume the acquired life cycle data is transmitted to the PSIMS over the Internet.*

Background: For the PS2 game console, medium or strong shocks can cause damage of the CD drive (misalignment of the laser). A defective CD-drive might start operation but stop after initial turns. In this case the maximum speed detected for the defective CD-drive is < 500 rpm.

Remote Diagnostics: In PSIMS, the dynamic data collected from unit 0719952J136 are displayed in Figure 9.25.

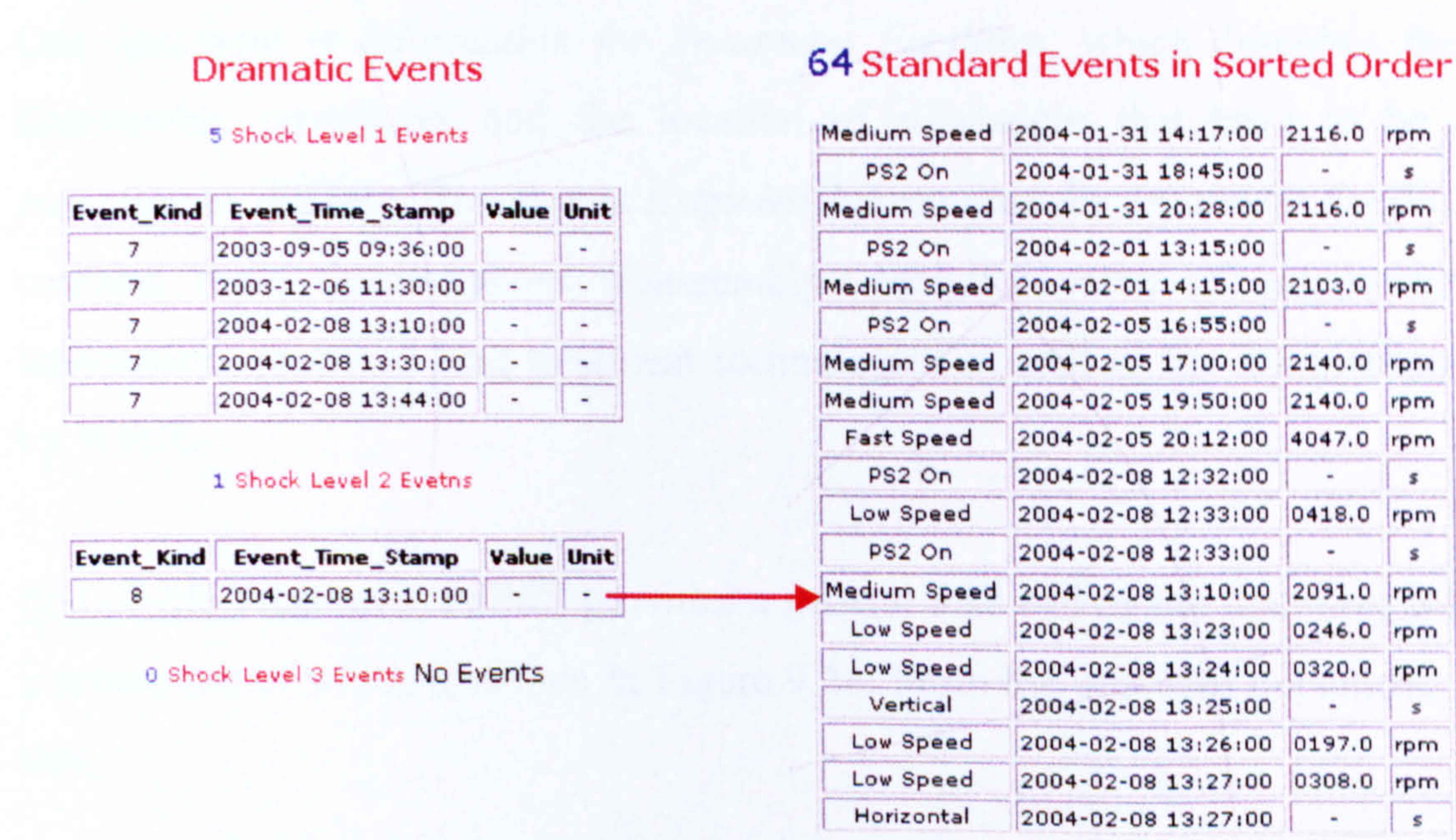


Figure 9.25 Remote diagnostics: A data display screenshot from PSIMS

In the left-hand table of dramatic events, it shows a shock occurred (event type 8) at 2004-02-08 13:10:00. The shock level 2 means this PS2 unit encountered a disturbance



moving from the vertical to a horizontal position. In the right-hand table of standard events, it can be noted that all recorded events related to CD-drive are *low speed after* 2004-02-08 13:10:00. This means that after the shock damage to the CD-drive has occurred, as each recorded speed is below the typical operation speeds (of ~ 2000 rpm). Analysis of the result from the recorded data can indicate to maintenance technicians that the CD drive might be faulty.

#### 9.4.3.2 Scenario 2 - End-of-Life Treatment

The PS2 PSS prototype can deliver end-of-life treatment service.

*A defective, end-of-life PS2 returned from the customer. How to dispose? In other words, which components/material/substance can be potentially reused? Which components /material/substance can be recycled?*

Background: PSIMS provides two end-of-life treatment documents (i.e. static data). One document is *Information for Treatment Facilities*, which describes the general disassembly procedures and the location of substances that have to be removed according to WEEE. The other is *Extended Information for Treatment Facilities* which contains more details about disassembly, potential value of components, reuse information, proposed best treatment technologies as well as the information required by WEEE.

End-of-life Treatment: PSIMS provides a general PS2 end-of-life treatment information, a screenshot of which is shown in Figure 9.26. From the provided document, we know that:

1) Material for recycling, which are identified as follows:

- Lower cover
- Upper cover
- ABS
- Shield



- 2) Items for potential reuse, which are identified as follows:
- CD unit
  - Fan
  - Valuable sensors from ELIMA board (i.e. IDU)
- 3) Items which can be removed and treated separately according to the legislation, which are identified as follows:
- Battery pack
  - PWB
  - Power block
  - Capacitors
  - Chassis

Sony International (Europe) GmbH

61	Battery	1	3	Remove and treat separately according to legislation
132	Shield (A) assy	1	302	Remove heat sheets for pure metal fraction
90	PWB	1	150	Remove and treat separately according to legislation
141	Shield	1	25	Material recycling
94	Upper cover	1	218	Material recycling For high quality ABS recycling remove bracket, Sony logo and two rubber feet
	Material / substance:	Classification:	Weight:	Suggested treatment:
	ABS		218	Material recycling
300	ELIMA board and external components	1	47	Remove and treat separately according to legislation, valuable sensors Potentially fit for reuse
301	Battery pack	1	45	Remove and treat separately according to legislation
91	CD unit	1	444	Potentially fit for reuse
63	Fan	1	38	Potentially fit for reuse
62	PWB	1	66	Remove and treat separately according to legislation
60	Power block	1	269	Remove and treat separately according to legislation
201	Capacitor	1	1	Remove from power board and treat separately according to legislation
202	Capacitor	2	1	Remove from power board and treat separately according to legislation
96	Chassis	1	170	Sticker, power button, bracket unit and LED-window have to be removed for pure ABS fraction.
	Material / substance:	Classification:	Weight:	Suggested treatment:
	ABS		170	Material recycling
71	Label	1	1	Remove label from chassis for pure ABS fraction

Figure 9.26 A screenshot of PS2 end-of-life treatment information



### 9.4.3.3 Scenario 3 - Use Pattern Analysis

The PS2 PSS prototype can deliver the use pattern analysis service<sup>12</sup>.

*The PS2 console can be operated in two different positions: horizontal or vertical. We like to know which mode in which country is preferred by customers. Knowing this can help “Marketing Department” put more customer-preferable PS2 into that country, and can help “Design & Development Department” enhance the product design.*

Background: Within a manufacturing enterprise, use pattern analysis can be provided to *Marketing Department* to increase the market share, and to *Design & Development Department* to enhance the product design. The use pattern analysis is carried out based on the actual occurred dynamic data instead of questionnaire data where data might not complete and accurate. Thus the accuracy of the use pattern analysis can be assured.

Use Pattern Analysis: 205 life cycle data for PS2 units are housed in PSIMS. In order to minimise the influence of the soak test/recharge records on the results, 22 units with less than eight playing cycles from the latter field trials and 5 units without any playing events are excluded from the evaluation. Thus the base for the analysis are 178 units which from 4 countries, namely, Germany, UK, Spain, and Italy.

Figure 9.27 shows the analysis result. This analysis was carried out by SONY based on the Excel spreadsheet generated by PSIMS. Among 178 units, there are 105 units which were played in horizontal position (> 90% horizontal operation) and 21 units which were played in vertical position (> 90% vertical operation). The result shows that about 60% of the users preferred the horizontal position, about 12 % of the users preferred the vertical position and 28% of the users utilised PS2 in both positions (ELIMA D25, 2004).

The analysis indicates that most customers from whichever country prefer putting the console in horizontal position (60%) when playing the game. This use pattern analysis

---

<sup>12</sup> As defined in previous chapters, analysis of use pattern is regarded as a service in this research study



service can suggest *Design & Development Department* to enhance the product design which is more appropriate for running at horizontal position. The analysis also points out that there is no horizontal to vertical mode change identified for majority of the units in the standard event history. This implies that most users just select an initial operation position depending on their local environment. Once the position is selected, it will not be changed afterwards anymore. Hence the chance of using the game console as a portable device is very small.

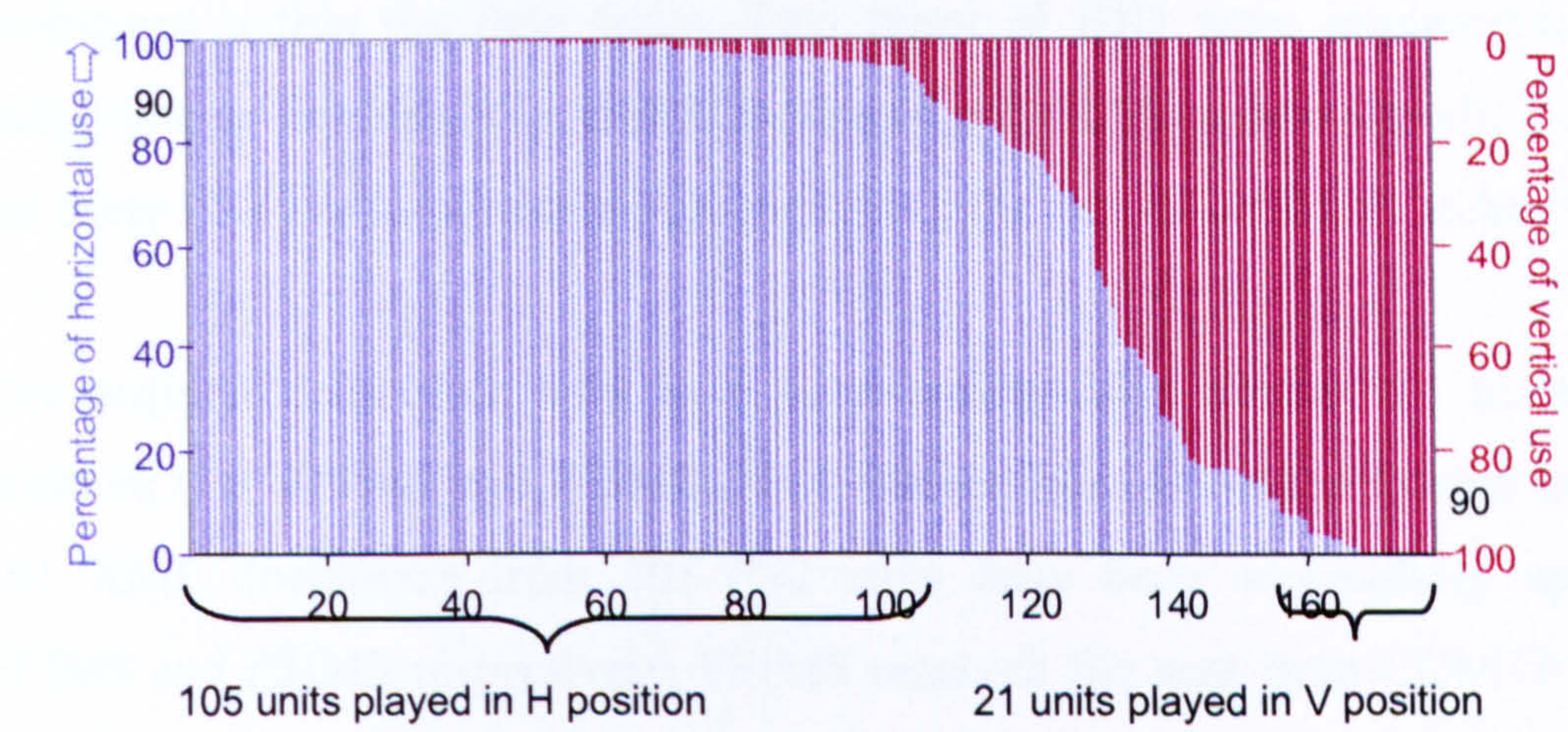


Figure 9.27 Preferred operation modes of PS2 consoles (ELIMA D25, 2004)

9.4.4 Summary of Test Case II

- 1) The PS2 PSS prototype has been developed by utilising the proposed PSS realisation methodology and associated toolkit.
- 2) An instance of the service enabler (i.e. PSIMS) has been created by employing the Framework and integrated into the field trial. The development of PS2 specific functionalities was described.
- 3) The following services have been delivered via the fridge/freezer PSS prototype.
  - Remote diagnostics
  - End-of-life treatment
  - Use pattern analysis



## 9.5 Appraisal of Test Results

It has been demonstrated that the actual test results can meet the expected results. Details are discussed as follows:

- 1) Two product-oriented PSS prototypes have demonstrated that they have capabilities to acquire the life cycle data. The implementations of IDU differ for the PSS prototypes within the field trials. Two types of IDU were implemented for the fridge/freezer field trial (i.e. EEPROM-based and GSM modem-based). The IDU for the Sony PS2 was implemented by employing the state-of-art RFID technology.
- 2) The acquired life cycle data have been successfully transmitted to the service enablers (i.e. FFIMS and PSIMS). 144 dataset from 89 fridge/freezer models and 301 XML documents from 205 PS2 units have been successfully uploaded to FFIMS and PSIMS respectively. FFIMS received file sent from COMTECH server almost every night from May, 2004 to December, 2004.
- 3) Services such as remote diagnostics, end-of-life product treatment, remote monitoring and use pattern analysis are delivered via the PSS prototypes.
- 4) The Framework was used to create two product-specific service enablers successfully. The two service enablers (i.e. FFIMS and PSIMS) created from the Framework meet the requirements specification.
  - Security Management - It supports role-session paradigm and 3-level security management. The security management reached the system functionality level and data level.
  - Data Communication - *File upload* utility has proved to be flexible and powerful.



- *Data Analysis* - The enabled services have proved that data analysis functionality (i.e. pre-process data and transform to Excel, on-line analysis) is effective and appropriate.
  - *Report generator* - It has been demonstrated working successfully. They can generate Excel spreadsheet and XML document.
  - *Scheduler* - It successfully scheduled the data extraction task. For example, the *scheduler* of FFIMS is scheduled to perform the data extraction task at 3:00am every day from May, 2004 to December, 2004.
  - *Notifier* - It successfully sent email out to the destinations. For example, three GSM data communication problems that happened in white goods field trial have all been captured by the *notifier*.
- 5) Two product-oriented PSS prototypes can help achieve the social, economic and environmental sustainability through the following delivered services:
- *Remote diagnostics service* - Remote diagnostics isolating the fault precisely and instantly before the visit can help reduce the maintenance cost.
  - *Remote monitoring service* - Remote monitoring product usage can ensure the product operates in a good condition and good environment so that its life span can be prolonged. It can also give suggestion of how to reduce energy consumption.
  - *End-of-life product treatment service* - Product end-of-life treatment can reduce the environment impact, and reduce cost through the reuse of components/material/substance.
  - *Use pattern analysis service* - Use pattern analysis can help manufacturer enhance the product design & development so that they can provide reliable and



efficient products to the society and gain more market share to create more economic values.

## 9.6 Summary

The proposed PSS realisation methodology and associated toolkit have been used in ELIMA field trials to build two product-oriented PSS prototypes. The evaluation of FFIMS and PSIMS and the demonstration of enabled services have shown that:

- The proposed methodology for the realising product-oriented PSS for consumer products is appropriate and effective.
- Tools, i.e. the Framework and supporting tools, are appropriate and effective to facilitate the development of product-oriented PSS.
- Enabled services can facilitate the realisation of sustainability.

Although the proposed methodology and tools have been evaluated through product-oriented PSS prototype, they also can be applied to build use-oriented PSS for consumer products to enable product rental or sharing service. For example, information about how much energy is consumed during a certain period by a certain user of a fridge/freezer can be obtained from life cycle data to form energy consumption bill. Technically, the availability of this energy consumption bill of individual customer can enable the manufacturer to provide fridge/freezer rental or sharing service. Hence the way in which the methodology and toolkit are used to build use-oriented PSS is the same as to build product-oriented PSS.

The main differences are discussed as follows:

- 1) Apart from delivering services such as remote diagnostics, remote monitoring, end-of-life treatment, etc, use-oriented PSS usually can deliver rental or sharing services, which usually are not delivered through product-oriented PSS.



- 2) As for use-oriented PSS, the manufacturer is the owner of the product, the life cycle data can be more easily to be acquired than that of product-oriented PSS.



## Chapter 10 Conclusions and Future Work

### 10.1 Conclusions from the Research Study

The primary aim of this research study was to investigate methods and tools for the realisation of PSS for consumer products. The literature survey revealed that present engineering methods and tools have not yet fully resolved the issue of how services can be delivered during a product life cycle. In PSS research, there has been very little work carried out to develop systematic and integrated engineering tools which can help manufacturers implement and realise PSS solutions in a practical way. While many aspects of the research work has required detail designs, the main strength of the work lies in the amalgamation and synthesis of a range of technologies to achieve practical applications for the benefit of manufacturers and users alike.

Hence in this research study,

- 1) An engineering methodology for realising product-oriented PSS and use-oriented PSS for consumer products has been developed. The proposed methodology has linked various aspects of technologies together and takes full advantage of the product life cycle data.

The complete realisation approach is described as follows:

- a) Design IDUs and embed into the products to acquire the dynamic life cycle data.
- b) Use the Framework to generate a service enabler skeleton
- c) Develop product-specific functionalities by using the class libraries and APIs provided by the Framework
- d) Plug the product-specific functionalities into the service enabler skeleton to formulate the product-specific service enabler
- e) Transmit the dynamic data and static data to the service enabler



- 2) The design of a service enabler requires a set of complex tools. A toolset has been specially developed for this purpose – called the Framework. It encompasses a set of components, class libraries and APIs, dynamic data structure schema, life-cycle database schema and runtime infrastructure. The Framework together with associated supporting tools has been extensively used in the development of the prototype IMSs.
- 3) The methodology and tools have been tested in ELIMA field trials. The methodology was employed to build two product-oriented PSS prototypes, and the Framework was used to create two actual instances of the service enabler, namely, FFIMS and PSIMS. The field trials have been used to demonstrate that the two PSS prototypes can deliver valuable service content. Further, proposed methodology and tools have been demonstrated to be effective and appropriate.

Main challenges encountered in the research study include:

- 1) This research study welds together many different aspects of ICT and engineering technologies to provide product-related services to various stakeholders involved. Therefore tasks such as to develop a user-friendly interface, to ensure the security of commercially sensitive product life data, to provide powerful statistical and analysis capabilities and to support different data acquisition and communication mechanisms are major challenges.
- 2) While the major tools are software based, the design of these tools requires a good understanding of engineering disciplines such as mechanical, electrical and manufacturing engineering. Furthermore, the development of the tools requires a solid basis in ICT which includes database and information management system, system analysis and design, Web-based technologies such as XML, Web services and various Web-based programming languages such as Java, PHP, etc.
- 3) A lot of effort in this research study has been used in the design and development of the specific functionalities to meet the field trial requirements.



For example, due to the inadequate resolution of the date & time in the recorded event data, a series of specific algorithms have been developed to tackle the problems caused by this limitation

- 4) The development of tools was carried out in close cooperation with two well-known industry partners: Sony and Merloni UK. This imposed a strong requirements relating to many aspects of the project management. For example:
  - A clear understanding of requirements specification details through liaison with technical staff.
  - Scheduling and prioritising of the agreed tasks and their execution.
  - Overall time management in relation to set deadlines.
- 5) The ELIMA consortium has imposed some constraints on the research study. For example, as COMTECH is a contractor responsible for lively collecting data through GSM modem from fridge/freezer, the lively acquired life cycle data has to go to the COMTECH first. COMTECH then packaged the data and transmitted them to service enabler at DMU on a daily basis. This has impact on the research and development of tools to support the live data transfer to service enabler.

## **10.2 Contribution to Knowledge**

The proposed methodology for realising PSS can help consumer goods manufacturers to design and develop next generation consumer products (e.g. sustainable smart home appliances) which are appropriate for new business models. It would open new horizons for product eco-designs which are sustainable and environment sensitive. Furthermore, it would supplement the discipline of service engineering by proposing a service enabling scheme to explore the delivery of product-related services.



The research study would open new dimensions for Product Lifecycle Management (PLM) through acquiring product life cycle data, closing the product information loop, and managing them in a systematic and integrated manner.

The research study contributed to the investigation of methods and tools for manufacturers to meet the on-coming legislation of WEEE and RoHS Directives. The research work provides a systematic and integrated way to manage the valuable product life cycle data so that manufacturer, recycler / refurbisher, etc can get timely and accurate information to help dispose the end-of-life product.

The research study also contributed to the investigation of methods to reduce energy consumption for consumer products. Managing product life cycle data in a systematic and integrated way can help manufacturers find and study the product use pattern. This would assist manufacturers to produce energy-efficient consumer products.

### **10.3 Recommendations and Future Work**

Although the developed product life cycle information management system provides data analytical capabilities, they can be much more comprehensive and powerful. The advance of new technologies will provide opportunities to improve and add power to the ICT such as data mining and other mathematical techniques. It is recommended that continued research and development should give more attention to incorporating data mining and knowledge base into the system to maximise the value of product life cycle data and provide seamless services.

The proposed PSS realisation methodology is mainly concerned with the delivery of product-oriented services and use-oriented services. How PSS can deliver pure result-oriented services (e.g. warmth-delivery instead of delivering heating devices) leading to a new business model (e.g. to have service instead of ownership) is recommended for further investigation.



The proposed methodology and tools are restricted to consumer products. However, there are other sectors in industry which can benefit from this methodology. For example, in the motor car industry, life cycle data generated during the use of the vehicle can be collected, stored, transmitted, managed and analysed by applying a PSS model. Similar ideas can be considered for aircraft industry. To some extent, the proposed methodology and tools can be extended to apply to most physical goods industry. Theoretically, the “intelligent” technologies such as sensor, Auto-Id, wireless communication, etc can be applied to any physical products (i.e. cup, table) to acquire their life cycle data and transmit them to the service enabler to enable services.

Due to the restriction of the ELIMA project and field trials, the acquisition and transmission of product life cycle data in a live manner has not been investigated in great depth. This may need further research and development.



## References

- Adriaans, P., Zantinge, D. (1996) *Data Mining*, Addison-Wesley, 1996.
- Apte, C. and Hong, S. (1996) "Predicting equity returns from securities data", in Fayyad, U., Piatetsky-Shapiro, G., Smythand, P. and Uthurusamy, R.(1996) *Advances in Knowledge Discovery and Data Mining*, AAAI Press /MIT Press, pp. 542–560.
- Arnold, K., Gosling, J., Holmes, D. (2000) *The Java Programming Language* 3<sup>rd</sup> edition, Addison Wesley.
- Astrahan, M.M., Blasgen, M.W., Chamberlin, D.D., et al (1976) "System R: relation aapproach to database management", *ACM transactions on Database Systems*, vol. 1, no.22, pp.97-137.
- Atzeni, P., Ceri, S., Paraboschi, S., Torlone, R. (1999) *Database Systems*, the McGraw-Hill Companies.
- Aurich, J.C., Fuchs, C. (2004) "Life cycle oriented technical PSS design", *Proceedings of SusProNet Conference on Product Service Systems: Practical Value*, 3&4, June 2004, Brussels, Belgium, pp.7-9.
- Aurich, J.C., Wagenknecht, C. (2003) "Model-based production planning for networked production systems", *Proceedings of 36<sup>th</sup> CIRP International Seminar on Manufacturing Systems*, Saarbrücken, Germany, pp. 447-453.
- Bakken, S. S., Aulbach, A., Schmid, E., Winstead, J., Wilson, L. T., Lerdorf, R., Zmievski, A., Ahto, J.(2004) *PHP Manual*, PHP Documentation Group, Available: <http://www.php.net/manual> (Accessed: 02, September, 2004).
- Becerra-Fernandez, I. (2001) "Searching for experts with expertise-locator knowledge management systems", *Proceedings of the Association for Computational Linguistics Conference (ACL 2001) Workshop on Human Language Technology and Knowledge Management*, July, 2001, Toulouse, France.
- Beck, K. (1999) *Extreme Programming Explained: Embrace Change*, Addison-Wesley.
- Bennett, S., McRobb, S., Farmer, R. (2002) *Object-Oriented Systems Analysis And Design Using UML*, 2<sup>nd</sup> edition, The McGraw-Hill Companies.
- Bodenhoefer, K., Schneider, A., Cock, T., Brooks, A., Sands, G., Allman, L., Simon, M., Chong, S. K., Yang, K. X., Delannoy, O., Catanese, B., Mueller, K., (2004) "Environmental life cycle information management and acquisition: first experiences



- and results from field trials”, *Proceedings of Conference Electronics Goes Green 2004+*, September 6-8, 2004, Berlin, Germany.
- Boehm, B.W.** (1988) “A spiral model of software development and enhancement”, in Thayer, R.H. (Ed.), *Tutorial: Software Engineering Project Management*, Los Alamitos, CA: IEEE Computer Society Press.
- Booch, G.** (1994) *Object-oriented Analysis and Design with Applications*, 2<sup>nd</sup> edition, Benjamin/Cummings, Redwood City, California.
- Bowers, M.R.** (1989) “Developing new services: improving the process makes it better”, *Journal of Service Marketing*, vol.3, no.1, pp.15-20.
- Brezet, H.** (1997) “Dynamics in eco-design practice”, *UNEP Industry and the Environment*, vol. 20, no. 1-2, pp. 21-24.
- Britton, C., Doake, J.,** (2000) *Object-Oriented Systems Development: A Gentle Introduction*, McGraw-Hill Publishing Company, Berkshire, SL6 2QL, UK.
- Brown, H. S., P. Vergragt, et al.** (2002) “Learning processes in bounded social experiments: the case of Mitka”, paper presented at *10th International Conference of the Greening of Industry Network (GIN)*, June 23-26, 2002, Gothenburg, Sweden.
- Brown, S.A.**(1997) *Revolution at the Checkout Counter: The Explosion of the Bar Code*, Harvard University Press
- Brundtland, G. H.** (1987) *Our Common Future: the World Commission on Environment and Development*, Oxford University Press.
- Bullinger, H., Fahnrich, K.P., Meiren, T.** (2003) “Service engineering-methodical development of new service products”, *International Journal of Production Economics*, 85, (2003), pp.275-287.
- Cambridge Workshop** (2004) *Product end-of-life Management*, 17 of November, 2004, Downing College, Cambridge, UK.
- Castro, E.** (2000) *HTML 4 for the World Wide Web*, 4<sup>th</sup> edition, Peachpit Press, 1249 Eighth Street, Berkeley, CA, 94710.
- ChartDirector** (2005) Available: <http://www.advsofteng.com/> (Accessed: 27, February, 2005).
- Chen, P.** (1976) “The Entity-Relationship model: toward a unified view of data”, *ACM Transactions on Database Systems*, vol.1, no.1, pp. 9-36.



- Chong, S. K., Moor, P. R., Wong, C. B., Pu, J.** (2004) "Equipment information management in connected homes", *Proceedings of Mechatronics International Conference*, August 30-1 September, 2004, Ankara, Turkey.
- CODASYL** (1975) *Data Base Task Group Report 1971*, Association for Computing Machinery.
- Codd, E. F.** (1970) "A relational model of data for large shared databanks", *Communications of the ACM*, vol.13, no. 6, pp. 377-387.
- Codd, E. F.** (1979) "Extending the relational model to capture more meaning", *ACM Transactions on Database Systems*, vol. 4, no. 4, pp. 397-434.
- Codd, E. F.** (1990) *The Relational Model for Database Management*, Addison-Wesley.
- Connolly, T.M. & Begg, C.E.** (2002) *Database Systems*, 3rd Edition, Addison-Wesley.
- Delen, D., Walker, G., Kadam, A.** (2004) "Predicting breast cancer survivability: a comparison of three data mining methods", *Artificial Intelligence in Medicine*, vol.34, no.2, pp.113-127.
- Desai, N** (2002) "Global Challenge, Global Opportunity", published by the *United Nations Department of Economic and Social Affairs for World Summit on the Sustainable Development*, Johannesburg, 26 August - 4 September, Available: <http://www.Johannesburgsummit.org> (Accessed:20, April, 2005).
- Douglass, B. P.** (1999) *Real-Time UML: Developing Efficient Objects for Embedded Systems*, Second Edition, Addison-Wesley.
- DTI update on WEEE & RoHS** (2005) Available: [http://uk.farnell.com/static/en/rohs/news/dti\\_update.html](http://uk.farnell.com/static/en/rohs/news/dti_update.html) (Accessed: 7, April, 2005).
- Durkin, J.**(1994) *Expert Systems Design and Development*, Macmillan Publishing Company.
- Edgett, S.** (1994) "The traits of successful new service development", *Journal of Services Marketing*, vol. 8, no.3, pp. 40-49.
- ELIMA** (2004) Available: <http://www.elima.org> (Accessed: 7, May, 2005).
- ELIMA D17** (2004) "Final report of ELIMA white goods field trial", *ELIMA Internal Document*.
- ELIMA D25** (2004) "Final report of ELIMA brown goods field trial", *ELIMA Internal Document*.



- EPC can be used to identify item using RFID** (2004) Available: <http://www.rfid.zebra.com/epc.htm> (Accessed: 17, April, 2005).
- ELIMA Seminar** (2005) *Product Life Cycle Technology - Results and Implications for Business Strategy*, 23 of February, 2005, Vienna, Austria, presentation slides available: <http://www.elima.org> (Accessed: 3, May, 2005).
- Elmasri, R., Navathe, S.B.** (2000) *Fundamentals of Database Systems*, 3<sup>rd</sup> edition, Addison-Wesley.
- e-Service on Demand** (2005) Available: <http://ehome.mrg.dmu.ac.uk/eservice.htm> (Accessed: 05, June, 2005).
- EuP Directives** (2003) Available: [http://uk.farnell.com/static/en/rohs/infobank/directives/eup\\_directive.html](http://uk.farnell.com/static/en/rohs/infobank/directives/eup_directive.html) (Accessed: 17, April, 2005).
- Fayad, M., Schmidt, D., Johnson, R.**, (1995) *Building Application Frameworks – Object Oriented Foundations of Framework Design*, Wiley Computer Publishing.
- Feigenbaum, E., Friedland, P. E., Johnson, B. B., Nii, H.P., Schorr, H., Shrobe, H.** (1993) *Knowledge-based System in Japan*, Available: <http://www.wtec.org/loyola/kb/toc.htm> (Accessed: 12, March, 2005).
- Fiksel, J.** (1996) *Design for Environment: Creating Eco-Efficient Products and Processes*, McGraw-Hill, pp. 513.
- Finkenzeller, K.** (2003) *RFID Handbook*, 2<sup>nd</sup> edition, Wiley & Sons Ltd. Electronic version available: <http://RFID-Handbook.com> (Accessed: 20, April, 2005).
- Fitzsimmons, J. A., Fitzsimmons, M. J.** (2000) *New Service Development: Creating Memorable Experiences*, Sage Publications, Inc., 2455 Teller Road, Thousand Oaks, California, 91320.
- Flannagan, D.** (1997) *JavaScript: The Definitive Guide*, O'Reilly.
- Gamma, E. et al** (1995) *Design Patterns*, Addison-Wesley.
- Ganz, W., Meiren T.** (2004) "Co-Design of Products and Services", *Proceedings of SusProNet Conference on Product Service Systems: Practical Value*, 3&4, June 2004, Brussels, Belgium, pp.21-22.
- Gilb, T.** (1988) *Principles of Software Engineering Management*, Wokingham: Addison-Wesley, 1988.



- Glass, G. (2000) *Applying Web Services to Applications*, Available:<http://www-106.ibm.com/developerworks/webservices/library/ws-er1.html> (Accessed: 09, September, 2004).
- Goedkoop, M., van Halen, C., te Riele, H., and Rommens, P. (1999) *Product Service Systems, Ecological and Economic Basics*, Pre consultants: The Netherlands, Available: <http://www.pre.nl/pss/default.htm> (Accessed: 07, May, 2004).
- Gosling, J., Joy, B., Steele, G., Branche, G. (2000) *The Java Language Specification*, Addison-Wesley.
- Graham, I. (2001) *Object-oriented Methods: Principles & Practice*, 3<sup>rd</sup> edition, Addison-Wesley.
- Grudzien, W. & Seliger, G. (2000) "LCU - a concept for lifelong product supervision in closed loop economy", *Proceedings of Conference Electronics Goes Green 2000+*, Berlin, 11-13, 09, 2000, Berlin, pp. 397-402.
- Haller, S., Hodges, S. (2002) *The need for a Universal Smart Sensor Network*, Auto-ID Center, Institute of Manufacturer, University of Cambridge, UK.
- Han, J., Kamber, M. (2001) *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publisher, California, US.
- Harrison, M., Moran, H., Brusey, J., McFarlane, D. (2003) *PML Server Development*, White Paper, Auto-ID Center, Available: <http://www.autoidlabs.org/whitepapers/Cambridge> (Accessed: 22, April, 2005).
- Hockerts, K. (1999) "Eco-efficient service innovation: increasing business-ecological efficiency of products and services", chapter in: M., Polonsky, *Greener Marketing: A Global Perspective on Greener Marketing Practice*, Greenleaf Publishing, Sheffield, UK, pp.95-108.
- Hockerts, K. & Weaver, N. (2001) *Are service systems worth our interest? Assessing the eco-efficiency of sustainable service systems*, INSEAD University, Fontainebleau, France.
- Horst, van der, T.J.J., P. Vergragt, S. Silvester (1999) "Sustainable roadmapping: new approaches for identifying product system innovations". In: *1999 Greening of Industry Network Conference Best Paper Proceedings*, Chapel Hill, USA, November 14-17, 1999, p. 1-17.



- Intranet Design Magazine** (2000): *PHP from an IT Manager's Perspective*, June 4, 2000. Available: [http://linuxtoday.com/news\\_story.php3?ltsn=2000-06-04-002-04-NW-HL-SW](http://linuxtoday.com/news_story.php3?ltsn=2000-06-04-002-04-NW-HL-SW) (Accessed:25, August, 2005).
- IPP** (2005) *EUROPA–Environment Integrated Product Policy*, Available: <http://europa.eu.int/comm/environment/ipp/home.htm> (Accessed: 17, April, 2005).
- Jacobson, I., Ericsson, M., Jacobson, A.** (1999) *The Unified Software Development Process*, Addison-Wesley.
- Jacobson, I., Christerson, M., Jonsson, P., Overgaard, G.** (1992) *Object-Oriented Software Engineering, A Use Case Driven Approach*, Workingham: Addison-Wesley.
- Jakarta POI** (2005) Available: [http://jakarta.apache.org/poi/?bcsi\\_scan\\_5CBBA20D813A2009=0](http://jakarta.apache.org/poi/?bcsi_scan_5CBBA20D813A2009=0) (Accessed: 25, February, 2005).
- Java JRE** (2005) Available: <http://java.sun.com/j2se/1.3/runtime.html> (Accessed: 14, April, 2005).
- Java Mail** (2005) Available: <http://java.sun.com/products/javamail/> (Accessed: 14, April, 2005).
- Jessup, L, Valacich, J.** (2003) *Information Systems Today*, Prentice Hall, New Jersey, US, pp.71.
- Jeswiet, J.** (2003), “A definition of life cycle engineering”, *Proceeding of 36<sup>th</sup> CIRP International Seminar on Manufacturing Systems*, June 03 – 05, 2003, Saarland University, Saarbrücken, Germany, pp.17-20.
- JUpload Applet** (2004) Available: <http://www.haller-systemservice.net/jupload/> (Accessed: 17, April, 2005).
- J2EE** (2004) Available: <http://java.sun.com/j2ee>, Sun Microsystems, Inc. (Accessed: 20, August, 2004).
- Kango, R., Moore, P. R., Pu, J.** (2003) “Smart appliances and product service systems”, awarded best paper in the controls and communication theme at *International Conference on Computer, Communication and Control Technologies (CCCT)*, 2003, July 31, August 1-2, 2003, Orlando, Florida, US.
- Kauffman, J. J., White, A. et al.** (1997). “From Solvents to Services: Restructuring Chemical Supplier Relationships to Achieve Environmental Excellence”, *Proceedings*



of *IEEE International Symposium on Electronics and the Environment*, San Francisco, California, 5-7, May, 1997, pp. 322-326.

Kiritsis, D (2004) "Ubiquitous product lifecycle management using product embedded information devices", keynote paper presented at *Intelligent Maintenance System (IMS) 2004 International Conference*, 15-17, July, 2004, Arles, France, Available: <http://www.promise.no> (Accessed:7, April, 2005).

Klausner, M, Grimm, M.W.M., Hendrickson, C. (1998a) "Sensor-based recording of use conditions for product take back", *IEEE Symposium on Electronics & the Environment*, New Jersey: IEEE, pp. 138-143.

Klausner, M, Grimm, M.W.M., Hendrickson, C. (1998b) "Reuse of electric motors in consumer products: design and analysis of an electronics data log", *Journal of Industry Econology*, vol.2, no. 2, pp. 89-102.

Klein, M., Methlie, L.B. (1990) *Expert Systems: A Decision Support Approach: with Applications in Management and Finance*, Addison-Wesley.

Kovalerchuck, B., Triantaphyllou, E., Ruiz, J., Torvik, V. and Vityaev, E.(2000) "The reliability issue of computer-aided breast cancer diagnosis", *Journal of Computers and Biomedical Research*, vol. 33, no., 4, pp. 296–313.

Kroenke, D. M. (1998) *Database Processing*, 3<sup>rd</sup> edition, Prentice Hall International.

Larman, C. (1998) *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design*, Prentice-Hall, Inc.

Le, S.Y., Liu, W.M., Maizel, J.V. (2001) "A data mining approach to discover unusual folding regions in genome sequence", *Knowledge-based Systems*, vol. 15, no. 4, pp.243-250.

Lee, R. C., Tepfenhart, W. M. (2002) *Practical Object-Oriented Development with UML and Java*, Pearson Education International, Upper Saddle River, NJ 07458.

Lian, J., Lai, X. M., Liu, Z. Q., Yao, F. S. (2002) "Application of data mining and process knowledge discovery in sheet metal assembly dimensional variation diagnosis", *Journal of Materials Processing Technology*, vol. 129, no.1-3, pp. 315-320.

Mandelbaum, A. (1998) *Service Engineering: Modelling, Analysis and Inference of Stochastic Service Networks*, minicourse, Faculty of Industrial Engineering and Management Technion, Haifa, Israel.



- Manzini, E. and Vezzoli, C.**, (2002) *Product-Service Systems and Sustainability*, United Nations Environment Program (UNEP), Division of Technology Industry and Economics (DTIE), Production and Consumption Branch, Paris.
- McFarlane, D.** (2002) *Auto-ID based Control-An Overview*, Auto-ID Centre, White Papers from Cambridge University. Available: <http://www.autoidlabs.org/whitepapers/CAM-AUTOID-WH-004.pdf> (Accessed: 9, April, 2005).
- McFarlane, D., Sarma, S., Chirn, J.L., Wong, C.Y., Ashton, K.** (2003) "Auto ID systems and intelligent manufacturing control", *Journal of Engineering Application of Artificial Intelligence*, Available: <http://www.sciencedirect.com> (Accessed: 9, April, 2005).
- McFarlane, D.** (2002) *Auto-ID Based Control – An Overview*, white paper, Auto-ID Center. Available: <http://www.autoidlabs.org/whitepapers/Cambridge> (Accessed: 9, April, 2005).
- Meijkamp, R.** (1994). "Service-Products, a Sustainable Approach? A Case Study on Call-a-Car in the Netherlands", *Eco-efficient services seminar at the Wuppertal Institute*. Wuppertal: September 18-19, 1994.
- Meijkamp, R.** (1998) "Changing consumer behaviour through eco-efficient services: an empirical study of car sharing in the Netherlands". In: *Business Strategy and the Environment*, vol. 7, pp. 234-243.
- Microsoft COM** (2005) Available: <http://www.microsoft.com/com/default.msp> (Accessed: 19, February, 2005).
- Microsoft Task Scheduler** (2005) Available: <http://www.microsoft.com/technet/prodtechnol/windows2000serv/evaluate/featfunc/taskschd.msp> (Accessed: 9, February, 2005).
- Middleware** (2004) Available: <http://www.sei.cmu.edu/str/descriptions/middleware.html> (Accessed: 19, January, 2005).
- Mont, O.** (2000) *Product-Service Systems*, AFR-REPORT 288, prepared for the Swedish Environmental Protection Agency, Stockholm, Sweden.
- Mont, O.** (2003) "Learning the lessons, paving the way", *Proceedings of 1<sup>st</sup> SusProNet Conference*, 5-6, June 2003, Amsterdam.



- Moore, P.R., Pu, J., Wong, C. B., Chong, S. K., Yang, X. Y.** (2003) “A Component-based development environment for environmental life-cycle information management systems for consumer products”, *International Conference on Computer, Communication and Control Technologies (CCCT)*, July 31, August 1-2, 2003, Orlando, Florida, US.
- Moore, P.R., Pu, J., Xie, C., Simon, M., Bee, G.** (2000) “Life cycle data acquisition methods and devices”, *Proceedings of the 7<sup>th</sup> Mechatronics Forum International Conference (Mechatronics 2000)*, 6<sup>th</sup> – 8<sup>th</sup> September, 2000, Atlanta, Georgia, USA.
- Mooney III, V. J., Blough, D. M.** (2002) “A Hardware-Software Real-Time Operating System Framework for SoCs”, *IEEE Design & Test*, vol.19, no. 6, pp. 44 – 51.
- Morelli, N.** (2004) “Developing new PSS: methodologies and operational tools”, *Proceedings of SusProNet Conference on Product Service Systems: Practical Value*, 3&4, June 2004, Brussels, Belgium, pp.44-47.
- Mueller, K.** (2003) “Generic Data Requirements and Information Feedback”, D10, *Deliverable Report of ELIMA Project (Internal)*.
- Mysql** (2005) Available: <http://www.mysql.com> .
- Netcraft Web Server Survey** (2005) Available: [http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html) (Accessed: 17, February, 2005).
- Norris, P., Coats, B.,** (2001) *Three-tier Web Application Development*, notes of E-Commence software module, De Montfort University, Leicester, UK.
- Océ: The environment.** Available: [http://www.oceusa.com/html/cp\\_envIRON.html](http://www.oceusa.com/html/cp_envIRON.html) (Accessed: 23, July, 2004).
- Pasetti, A, Pree, W, Terraillon, L., Overbeek, T. V.** (2001) “An object-oriented component-based framework for on-board software”, *Data Systems in Aerospace, DASIA*, 28 May - 1 June 2001, Nice, France.
- PhpMyAdmin** (2005) Available: <http://www.phpmyadmin.net/>.
- Pi-Net** (2002) *Pi-Net PSS Information Network* Available: <http://www.pss-info.com> (Accessed: 27, July, 2004).
- PROMISE** (2005) *Product Lifecycle Management and Information tracking using Smart Embedded Systems*, Available: <http://www.promise.no> (Accessed: 7, April, 2005).



- Pyle, D. (2003) *Business Modelling and Data Mining*, Morgan Kaufmann Publisher, California, US.
- Ramaswamy, R. (1996) *Design and Management of Service Processes*, Addison-Wesley.
- Online RFID Handbook (2005) Available: <http://RFID-Handbook.com> (Accessed: 20, April, 2005)
- Rolland, F.D. (1998) *The Essence of Database*, Prentice Hall.
- Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W. (1991) *Object-Oriented Modeling and Design*, Prentice-Hall.
- Sarma, S.E., Weis, S.A., Engels, D.W. (2003) *RFID Systems, Security & Privacy Implications*, Auto-ID Centre, Massachusetts Institute of Technology, US, Available: <http://www.autoidlabs.com/researcharchive> (Accessed: 17, April, 2005).
- Scheidt, L-G & Zong, S (1994) "An approach to achieve reusability of electronic modules", in *Proceedings of IEEE International Symposium on Electronics & the Environment. IEEE*, San Francisco, US, pp.331-336.
- Schrader, U. (1999) "Customer acceptance of eco-efficient services: a German perspective", *Greener Management International*, vol., 25, pp.105-121.
- Seliger, G., Grudzien, W., Zaidi, H. N. (1999) "New methods of product data provision for a simplified disassembly" in *7<sup>th</sup> CIRP Seminar on Life Cycle Engineering*. Kingston, Ontario, Canada.
- Sell, P. S. (1985) *Expert systems: a practical introduction*, Basingstoke: Macmillan.
- Shipman, D.W. (1981) "The functional data model and the language DAPLEX", *Transactions on Database Systems*, vol. 6, no.1, pp.140-173.
- Short, K. (1997) *Component Based Development and Object Modelling*, Sterling Software.
- Silberschatz, A., Stonebraker, M., Ullman, J. (1996) *Database Research: Achievements and Opportunities in 21<sup>st</sup> Century*, Technical Report CS-TR-95-1563, Department of Computer Science, Stanford University, Stanford, US.
- Simon, M., Moore, P. R. and Pu, J. (1998) "The WHITEBOX-capturing and using product life cycle data", *Proceedings 7<sup>th</sup> CIRP Life Cycle Design '98 Seminar*, Stockholm, Sweden, 16-18<sup>th</sup> September, pp.161-170.



- Simon, M., Allman, L., Moore, P.R., Yang, K. X., Cock, T. (2004) "Product design informed by life cycle information: initial results from ELIMA", *2<sup>nd</sup> International Conference on Manufacturing Research (ICMR2004)*, September, 7-9, 2004, Sheffield Hallam University, UK.
- Simon, M., Bee, G., Moore, P., Pu, J., Xie, C. (2000) "Life cycle data acquisition unit-design, implementation, economics and environmental benefits", *IEEE International Symposium on Electronics & the Environment*, May, 2000, San Francisco, US, pp: 284-289.
- Sission, D. (2002) *A Thoughtful Approach to Web Site Quality*, Available: <http://www.philosophe.com/reference/terms.html> (Accessed:24, March, 2005).
- Stevels, A.L.N. (1997) "Moving companies towards sustainability through eco-design", In: *The Journal of Sustainable Product Design*, issue 3, pp. 47-55.
- Sturm, J. (1999) *VB6 UML Design and Development*, Wrox Press Ltd.
- SusProNet (2003) *Network on Sustainable Product Service System Development*, Available: [www.suspronet.org](http://www.suspronet.org) (Accessed: 24, May, 2005).
- Tischner,U. (2004) "SusProNet results need area offices", *Suspronet Dissemination Event*, 13/10/2004, Brussels, Belgium
- Thearling, K. (2004) Available: <http://www.thearling.com> (Accessed: 13, November, 2004).
- The EPCglobal Network (2004). Available: [http://www.epcglobalinc.org/news/EPCglobal\\_Network\\_Overview\\_10072004.pdf](http://www.epcglobalinc.org/news/EPCglobal_Network_Overview_10072004.pdf) (Accessed: 23, March, 2005).
- Tomiyama, T. (1997) "A manufacturing paradigm toward the 21<sup>st</sup> century", *Integrated Computer Aided Engineering*, vol.4, pp.159-178.
- Tomiyama, T., Medland, A. J., Vergeest, J. S. M.(2000) "Knowledge intensive engineering towards sustainable products with high knowledge and service contents", *TMCE 2000,Third International Symposium on Tools and Methods of Competitive Engineering*, April,18-20, Delft University Press, Delft, the Netherlands, pp.55-67.
- Tomiyama, T. (2003) *Service CAD*, proceedings of *1<sup>st</sup> SusProNet Conference*, Amsterdam, 5-6 June, 2003.



**TopXML** (2004) Available:

[http://www.topxml.com/php\\_xml\\_dom/how\\_to\\_install\\_php\\_in\\_different\\_environment.asp](http://www.topxml.com/php_xml_dom/how_to_install_php_in_different_environment.asp) (Accessed: 13, February, 2005).

**Tukker, A.** (2003) "CO2 reduction from household Consumption by PSS", *Sustainable Consumption Workshop*, 12-13, December, Tokyo

**Tukker, A.** (2004) "Product-services as a means to reach sustainable consumption", *3<sup>rd</sup> workshop on Sustainable Consumption and Production*, 21-22, October, 2004, Tokyo.

**Udoka, S. J.** (1992) "The role of automatic identification in the Computer Integrated Manufacturing (CIM) architecture", *Computers & Industrial Engineering*, vol. 23, no. 1-4, pp.1-5.

**UK Government Sustainable Development** (2004) Available:

<http://www.sustainable-development.gov.uk/index.htm> (Accessed: 15, July, 2004).

**UNCED** (1992) *Agenda 21*, result of the United Nations Conference on Environment and Development (UNCED), Rio de Janeiro, 3-14, June, 1992.

**UNEP** (1994) *Elements for Policies for Sustainable Consumption*, UNEP (United Nations Environment Program) Cleaner Production Programme.

**UNEP & PSS** (2001) *UNEP and Product Service Systems* Available: <http://www.uneptie.org/pc/sustain/design/design.htm> (Accessed: 24, July, 2004).

**Vawter, C., Roman, E.** (2001) *J2EE vs. Microsoft .NET*, prepared for Sun Microsystems, Inc. Available: <http://www.theserverside.com/articles/pdf/J2EE-vs-DotNET.pdf> (Accessed: 17, August, 2004).

**Vergragt, P. J.** (2001) "How to achieve system innovation: some lessons from visioning and back-casting project", *Proceedings of 9<sup>th</sup> Greening of Industry*, 21 - 24 January, Bangkok, Thailand, 2001.

**Vidgen, R., Avison, A., Wood, D., Harper, T.W.** (2002) *Developing Web Information Systems*, Butterworth-Heinemann.

**Vlist, E.** (2002) *XML Schema*, O'Reilly & Associates, Inc.

**Votta, T.** (2001) Servicizing, lecture to the *6<sup>th</sup> Towards Sustainable Product Design Conference*, Amsterdam, October 2001.

**WBSCD** (1998) *Cleaner Production and Eco-Efficiency-Complementary Approaches to Sustainable Development*, the World Business Council for Sustainable Development and United Nations Environment Program-Industry and Environment, Paris, 1998.



**WEEE & RoHS Directive –DTI Website**, available:

<http://www.dti.gov.uk/sustainability/weee/index.html> (Accessed: 14, September, 2004).

**Welford, R.J.** (1996) *Corporate Environmental Management: Systems and Strategies*, Earthscan (publisher), London.

**Wheelwright, S.C., Clark, K.B.** (1992) *Revolutionizing Product Development – Quantum Leaps in Speed, Efficiency, and Quality*, a Free Press, a Division of Simon & Schuster Inc., New York.

**White, A. L., Stoughton, M., Feng, L.** (1999) *Servicizing: The Quiet Transition to Extended Product Responsibility*, Tellus Institute for Resource and Environmental strategies, Boston, US.

**Wirfs\_Brick, R., Wilkerson, B. Wiener, L.** (1990) *Designing Object-oriented Software*, Prentice-Hall.

**Wind, J., Mahajan, V.** (1997) “Issues and opportunities in new product development: an introduction to the special issue”, *Journal of Marketing Research*, vol.34, no.1-12.

**Windows Comparison** (2004) Available: <http://www.dedipower.com/d/windows-comparison/>, Asuk Creative Limited (Accessed: 15, May, 2005).

**Wong, M** (2003) *Product Service Systems in Consumer Goods Industry*, PHD thesis, Cambridge University, UK.

**W3C** (1999c) *XML Path Language (XPath) 1.0*, World Wide Web Consortium, 16 November, 1999, Available: <http://www.w3c.org/TR/xpath> (Accessed: 8 September, 2004).

**W3C** (2000b) *XML 1.0*, 2<sup>nd</sup> edition, World Wide Web Consortium, 6 October 2000, Available: <http://www.w3c.org/TR/REC-xml-20001006> (Accessed: 8 September, 2004).

**W3C** (2000d) *XML Pointer language (XPointer) 1.0*, World Wide Web Consortium, 7 June 2000, Available: <http://www.w3c.org/TR/xptr> (Accessed: 8 September, 2004).

**W3C** (2000f) *XML Schema Part 0: Primer*, World Wide Web Consortium, 24 October 2000, Available: <http://www.w3c.org/TR/xmlschema-0> (Accessed: 8 September, 2004).

**Zanakis, S.H., Becerra-Fernandez, I.** (2002) “Competitiveness of nations: a knowledge discovery examination”, *European Journal of Operational Research*, vol. 166, no. 1, pp.185-211.

**Zend Engine** (2004) Available: <http://www.zend.com/store/products/zend-engine-in-depth.php> (Accessed: 12, September, 2004).



**Züehlke Engineering AG** (2002) *Unified Software Development Process*. Available : <http://www.zuehlke.com/> (Accessed : 01 March, 2005)



# Appendix A: Class Libraries and APIs

## A.1 Classes and APIs in Java

### A.1.1 dbms.ElimadbManager.Elimadb

dbms.ElimadbManager

#### Class Elimadb

java.lang.Object

└ dbms.ElimadbManager.Elimadb

public class **Elimadb**

extends java.lang.Object

Constructor Summary	
<a href="#">Elimadb()</a>	
Method Summary	
void	<a href="#">closeConnection()</a>
void	<a href="#">closeDbConnection()</a>
java.lang.String[]	<a href="#">convertTo1DDataset</a> (java.lang.String queryString)
void	<a href="#">convertTo2DDataset</a> (java.lang.String queryString)
void	<a href="#">deleteData</a> (java.lang.String queryString)
java.lang.String[][]	<a href="#">getBn2DDataset</a> (java.lang.String queryString)



java.lang.String[]	<u>getResultSetColName</u> (java.lang.String queryString)
int	<u>getResultSetColNo</u> (java.lang.String queryString)
int	<u>getResultSetRowNo</u> (java.lang.String queryString)
boolean	<u>isResultSetEmpty</u> (java.lang.String queryString)
java.sql.Statement	<u>mySqlDbConnection</u> ()
java.sql.ResultSet	<u>retrieveData</u> (java.lang.String queryString)
java.lang.String	<u>retrieveSpecifiedItem</u> (java.lang.String queryString)
void	<u>serializeData</u> (java.lang.String queryString)

<b>Methods inherited from class java.lang.Object</b>
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait



A.1.2 Elimacore.ReportGenerator.ExcelGenerator

elimacore.ReportGenerator

Class ExcelGenerator

java.lang.Object

└─elimacore.ReportGenerator.ExcelGenerator

public class ExcelGenerator extends java.lang.Object

Constructor Summary

[ExcelGenerator\(\)](#)

Method Summary

void	<a href="#">createExcel</a> (java.lang.String[][] argDataSet, java.lang.String[] argDataHead,int argRowNo, int argColNo,java.lang.String[] argDataUnit)
java.lang.String[]	<a href="#">getBnDataHead</a> ()
java.lang.String[][]	<a href="#">getBnDataSet</a> ()
java.lang.String[]	<a href="#">getBnDataUnit</a> ()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait



A.1.3 externalworld.Configurator.elimadatadefinition.DynamicDataField

externalworld.Configurator.elimadatadefinition

Class DynamicDataField

java.lang.Object

└ externalworld.Configurator.elimadatadefinition.DynamicDataField

public class **DynamicDataField** extends java.lang.Object

Constructor Summary

[DynamicDataField\(\)](#)

Method Summary

int	<a href="#">getBnFractionDigitForDay()</a>
int	<a href="#">getBnFractionDigitForMin()</a>
int	<a href="#">getCsvFileLineNo()</a>
int	<a href="#">getDataFieldNumber()</a>
java.lang.String	<a href="#">getDataStorePath()</a>
java.lang.String	<a href="#">getDbPassword()</a>
java.lang.String	<a href="#">getDbUser()</a>
java.lang.String[]	<a href="#">getDynamicDataField_EEPROM()</a>



java.lang.String[]	<a href="#"><u>getDynDataInGroup</u></a> (int argDataGroupId)
java.lang.String	<a href="#"><u>getDynDataQueryString</u></a> ()
java.lang.String	<a href="#"><u>getDynDataQueryStringInGroup</u></a> (int argDataGroupId)
java.lang.String[]	<a href="#"><u>getDynDataTbColName</u></a> ()
java.lang.String[]	<a href="#"><u>getDynDataUnit</u></a> ()
java.lang.String[]	<a href="#"><u>getDynDataUnitInGroup</u></a> (int argDataGroupId)
int	<a href="#"><u>getExcelCellWidth</u></a> ()
java.lang.String	<a href="#"><u>getExcelExportPath</u></a> ()
java.lang.String	<a href="#"><u>getExcelFileSavePath</u></a> ()
int	<a href="#"><u>getExcelTextFormatColNo</u></a> ()
java.lang.String	<a href="#"><u>getFileFolder</u></a> ()
java.lang.String	<a href="#"><u>getInsertString</u></a> ()
java.lang.String	<a href="#"><u>getProductName</u></a> ()
java.lang.String	<a href="#"><u>getXmlFileSavePath</u></a> ()
java.lang.String[]	<a href="#"><u>retrieveDataFromStringTo1DArray</u></a> (java.lang.String argOneDString)



	void <u><a href="#">setDataFieldNumber</a></u> (int argDataFieldNumber)
	void <u><a href="#">setDynamicDataField_EEPROM</a></u> ()
	void <u><a href="#">setDynDataTbColName</a></u> ()
	void <u><a href="#">setDynDataUnit</a></u> ()
java.lang.String[]	<u><a href="#">spiltStringByToken</a></u> (java.lang.String argString, java.lang.String token)

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait



A.1.4 externalworld.ElimaDataSource.ElimaFile

externalworld.ElimaDataSource.elimafile

Class ElimaFile

java.lang.Object  
└ java.io.File  
└ externalworld.ElimaDataSource.elimafile.ElimaFile

All Implemented Interfaces:

java.lang.Comparable, java.io.Serializable

public class **ElimaFile** extends java.io.File

See Also:

[Serialized Form](#)

Constructor Summary	
<a href="#">ElimaFile</a>	(java.lang.String pathName)
Method Summary	
void	<a href="#">closeExcelConnection</a> ()
void	<a href="#">closeExcelStatement</a> ()
int	<a href="#">getApplianceId</a> ()
int	<a href="#">getApplianceTypeId</a> ()
java.lang.String	<a href="#">getDataCollectDate</a> ()
java.lang.String	<a href="#">getDataCollectTime</a> ()



java.lang.String	<u><a href="#">getFileExtension()</a></u>
java.lang.String	<u><a href="#">getFileNameOnly()</a></u>
int	<u><a href="#">getFileNameStringLength()</a></u>
java.lang.String	<u><a href="#">getProductColorLeter()</a></u>
java.lang.String	<u><a href="#">getProductDataType()</a></u>
java.lang.String	<u><a href="#">getProductSerialNo()</a></u>
java.lang.String	<u><a href="#">getProductType()</a></u>
boolean	<u><a href="#">isAccessDocument()</a></u>
boolean	<u><a href="#">isApplianceExistedInDb()</a></u>
boolean	<u><a href="#">isApplianceTypeExistedInDb()</a></u>
boolean	<u><a href="#">isExcelDocument()</a></u>
boolean	<u><a href="#">isTextDocument()</a></u>
boolean	<u><a href="#">isXmlDocument()</a></u>
java.sql.Statement	<u><a href="#">openAccessFile()</a></u>
java.sql.Statement	<u><a href="#">openExcelFile()</a></u>



Methods inherited from class java.io.File				
canRead,	canWrite,	compareTo,	compareTo,	createNewFile,
createTempFile,	createTempFile,	delete,	deleteOnExit,	equals,
exists,	getAbsolutePath,	getCanonicalFile,	getCanonicalPath,	getName,
getParent,	getParentFile,	getPath,	hashCode,	isAbsolute,
isDirectory,	isFile,	isHidden,	lastModified,	length,
list,	list,	listFiles,	listFiles,	listFiles,
listRoots,	mkdir,	makedirs,	renameTo,	setLastModified,
setReadOnly,	toString,	toURI,	toURL,	

Methods inherited from class java.lang.Object
clone, finalize, getClass, notify, notifyAll, wait, wait, wait



A.1.5 externalworld.ElimaDataSource.elimafile.ElimaFileFolder

Class ElimaFileFolder

java.lang.Object  
└ java.io.File  
└ externalworld.ElimaDataSource.elimafile.ElimaFileFolder

All Implemented Interfaces:

java.lang.Comparable, java.io.Serializable

public class **ElimaFileFolder** extends java.io.File

See Also:

[Serialized Form](#)

Field Summary	
java.lang.String	<a href="#">elimaFileFolder</a>
java.lang.String	<a href="#">elimaFolder</a>
java.lang.String[]	<a href="#">fileAbsolutePathList</a>
Fields inherited from class java.io.File	
pathSeparator, pathSeparatorChar, separator, separatorChar	

Constructor Summary	
<a href="#">ElimaFileFolder</a>	(java.lang.String folderName)
Method Summary	
java.lang.String[]	<a href="#">getFileAbsolutePathList</a> ()



java.lang.String	<a href="#"><u>getFileFolderAbsolutePath()</u></a>
java.lang.String[]	<a href="#"><u>getFileNameList()</u></a>

Learn about the properties and operations related to MySQL database.

Methods inherited from class java.io.File
canRead, canWrite, compareTo, compareTo, createNewFile, createTempFile, createTempFile, delete, deleteOnExit, equals, exists, getAbsolutePath, getAbsolutePath, getCanonicalFile, getCanonicalPath, getName, getParent, getParentFile, getPath, hashCode, isAbsolute, isDirectory, isFile, isHidden, lastModified, length, list, list, listFiles, listFiles, listFiles, listRoots, mkdir, mkdirs, renameTo, setLastModified, setReadOnly, toString, toURI, toURL

Methods inherited from class java.lang.Object
clone, finalize, getClass, notify, notifyAll, wait, wait, wait



## A.2 Classes and APIs in PHP

### A.2.1 MysqlManager Class

# MysqlManager

Encapsulate the properties and operations related to MySQL database.

**private class MysqlManager**

## Private Method Summary

void [connectToDb\( \\$host, \\$user, \\$password, \\$dbName\)](#)

To connect to database

void [selectDb\( \\$dbName\)](#)

To select a database

void [QueryItem\( \\$query\)](#)

To get query result which contains only one item

void [Query\( \\$query\)](#)

To get a query result

void [get2DDataSet\( \\$query\)](#)

To get a query result presented in two-dimensional array



void get1DDataset( \$query)

To get a query result presented in one-dimensional array

void QueryRow( \$query)

To get row number of the query result

void Insert( \$query)

To insert data into database

void Update( \$query)

To modify data in database

void Delete( \$query)

To delete data from database

void getDbHostName()

To get host name of the database server

void setDbUserName( \$argUser)

To set database user name



void **setDbPassword( \$argPassword)**

To set database password

void **setDatabaseName( \$argDbName)**

To set database name

void **getDatabaseName()**

To get database name

void **setDbTableName( \$argDbTableName)**

To set table name

void **getDbTableName()**

To get table name

void **MySql\_ErrorMsg( \$msg)**

To capture error message



# TableGenerator

To encapsulate operations related to HTML table generation

private class TableGenerator

## Private Method Summary

void displayDynData( \$applianceId, \$displayTimePeriod, \$dynDataList)

To display data in HTML table

void createTable( \$argDataSet, \$argDataHead, \$argRowNo, \$argColNo, \$argDataUnit)

To create a HTML table with data unit

void createTableWithoutUnit( \$argDataSet, \$argDataHead, \$argRowNo, \$argColNo)

To create a HTML table without data unit

void createTableWithCheckBox( \$argDataSet, \$argDataHead, \$argRowNo, \$argColNo, \$formActionName, \$urlRoleName)

To create a HTML table with check box



void getColNo()

To get column number

void getRowNo()

To get row number

void transfer2DArrayToString( \$twoDArray, \$rowNo, \$colNo)

To transform data in two-dimensional array into concatenated string

void transfer1DArrayToString( \$oneDArray, \$len)

To transform data in one-dimensional array into concatenated string

## Private Field Summary

String \$oneDString

String \$twoDString



## A.2.3 SecurityManagement Class

Integer ScolNo

private class SecurityManagement

### Public Method Summary

Integer SrowNo

void SecurityManagement()

### Private Method Summary

array getViewableDynDataListInDescription( SargUserGroupNo)

Return a dynamic data list which is accessible by the specified user group No.

array getViewableDynDataListID( SargUserGroupNo)

Return a dynamic data list ID which is accessible by the specified user group No.

string getListTokenizer(array SargArray)

Return a string tokenizer delimited by comma

### Private Field Summary

String getViewableDynDataListInDescription



# SecurityManagement

private class SecurityManagement

## Public Method Summary

void [SecurityManagement\(\)](#)

## Private Method Summary

array [getViewableDynDataListInDescription\( \\$argUserGroupNo\)](#)

Return a dynamic data list which is accessible by the specified user group No.

array [getViewableDynDataListId\( \\$argUserGroupNo\)](#)

Return a dynamic data list ID which is accessible by the specified user group No.

string [buildStringtokenizer\(array \\$argArray\)](#)

Return a string tokenizer delimited by comma

## Private Field Summary

String [\\$viewableDynDataListInDescription](#)



# ElimaAnalysisFunctions

To encapsulate standard analysis functions.

private class ElimaAnalysisFunctions

## Public Method Summary

void [ElimaAnalysisFunctions\(\)](#)

Constructor

## Private Method Summary

void [StdDev\(array \\$argArrayData\)](#)

Standard deviation

void [Correl\( \\$argArrayData1, \\$argArrayData2\)](#)

Correlation

void [Pearson\( \\$argArrayData1, \\$argArrayData2\)](#)

Pearson Correlation



void [ftest\( \\$argArrayData1, \\$argArrayData2\)](#)

F-Test

void [maxValue\( \\$argArrayData1\)](#)

Maximum value

void [minValue\( \\$argArrayData1\)](#)

Minimum value

void [average\( \\$argArrayData1\)](#)

Average

void [geomean\( \\$argArrayData1\)](#)

Geometric mean

void [harmean\( \\$argArrayData1\)](#)

Harmolic mean

void [kurtosis\( \\$argArrayData1\)](#)

Kurtosis



## ChartGenerator

void [median\( \\$argArrayData1\)](#)

Median

## Private Method Summary

void [skew\( \\$argArrayData1\)](#)

Skewness

void [devsq\( \\$argArrayData1\)](#)

Sum of squares of deviations

void [frequency\( \\$argData, \\$argBin\)](#)

Frequency distribution



A.2.5 ChartGenerator Class

# ChartGenerator

private class ChartGenerator

## Private Method Summary

void [createMultipleLineChart\( \\$data0, \\$data1, \\$data2, \\$data3, \\$data4, \\$labels, \\$len, \\$chartTitle, \\$x title, \\$y title, \\$line1, \\$line2, \\$line3, \\$line4, \\$line5\)](#)

To create multiple-line chart

void [createBarChart\( \\$data, \\$labels, \\$len, \\$chartTitle, \\$x title, \\$y title\)](#)

To create bar chart

void [createComboChart\( \\$data, \\$labels, \\$len, \\$chartTitle, \\$x title, \\$y title\)](#)

To create combo chart

void [createLineChart\( \\$data, \\$labels, \\$len, \\$chartTitle, \\$x title, \\$y title\)](#)

To create line chart

void [createScatterChart\( \\$data, \\$labels, \\$len, \\$chartTitle, \\$x title, \\$y title\)](#)

To create scatter chart



A.2.6 DropDownMenuBuilder Class

# DropDownMenuBuilder

To encapsulate properties and operations related to drop-down menu

**private class DropDownMenuBuilder**

## Private Method Summary

void [createDropDownMenu\( \\$menuListNo, \\$menuListItem,  
\\$dropDownMenuName, \\$defaultCategory\)](#)

To create drop-down menu with index

void [createDropDownMenu NoIndex\(\\$menuListNo,\\$menuListItem,  
dropDownMenuName\)](#)

To create drop-down menu without index

void [getListItemFromDb\( \\$queryString\)](#)

To get list items from database

void [getListItemNumberFromDb\( \\$queryString\)](#)

To get list item number from database



# Private Field Summary

## DynDataGrouping

Integer [\\$menuListNo](#)

To encapsulate properties and operations related to the group of dynamic data

private class DynDataGrouping

Array [\\$menuItem](#)

## Summary

void [initGroupData\(int menuListNo\)](#)

To group data

void [setDynamicDataDescription\(String desc\)](#)

To set dynamic data description to the group

void [setDynamicGroup\(\)](#)

void [setDynamicDataGroup\(\)](#)

To set dynamic data group

void [setDynamicDataMenuListNo\(\)](#)

To set dynamic data menu list no

void [setDynamicDataMenuListNo\(\)](#)

To set dynamic data menu list no



# DynDataGrouping

To encapsulate properties and operations related to the group of dynamic data

**private class DynDataGrouping**

## Private Method Summary

void [buildGroupData\( \\$argDataGroupNo\)](#)

To group data

void [getDynDataDescriptionInGroup\(\)](#)

To get dynamic data description in the group

[getDynDataInGroup\(\)](#)

void To get dynamic data in group

void [getDynDataUnitInGroup\(\)](#)

To get dynamic data unit in group

void [getDynDataNoInGroup\(\)](#)

To get dynamic data number in group



# Private Field Summary

Field Name	Description
Array <a href="#"><u>\$dynDataInGroup</u></a>	Used to support the creation of database
<a href="#"><u>InsertDynamicDataIn.java</u></a>	Used to insert the product-specific dynamic data into defined by the developer into the database
Array <a href="#"><u>\$dynDataUnitInGroup</u></a>	Used to automate the task of extracting data from files and saving them into the database in the case of data is lost or damaged.
<a href="#"><u>InsertSpecificFunctionality.java</u></a>	Used to plug a product-specific functionality into the service enabler skeleton.
<a href="#"><u>FunctionalityReplacement.java</u></a>	Used to replace the default functionality provided by service enabler framework
<a href="#"><u>HexDecimalConverter.java</u></a>	Used for converting Hexadecimal reading into decimal
<a href="#"><u>EmailTemplateMail.java</u></a>	Used to send out an email



# Appendix B: Supporting Tools

Tool Name	Description
createDb.bat	Used to support the creation of database
InsertDynamicDataList.java	Used to insert the product-specific dynamic data item defined by the developer into the database
ElimaDataUpdate.java	Used to automate the task of extracting data from files and saving them into the database in the case of data is lost or damaged.
InsertSpecificFunctionality.java	Used to plug a product-specific functionality into the service enabler skeleton.
FunctionalityReplacement.java	Used to replace the default functionality provided by service enabler framework
HexDecimalConverter.java	Used for converting Hexadecimal reading into decimal
ElimaSimpleMail.java	Used to send out an email



## Appendix C: List of Publications

1. **Yang, K. X., Moore, P. R., Wong, C.B., Pu, J., Chong, S. K. and Simon, M** (2004) "A methodology for implementing product-service systems for consumer products", *Proceeding of SusProNet Internal Conference, Product Service Systems: Practical Value*, 3&4 June 2004, Brussels, Belgium. Available: <http://www.suspronet.org> .
2. **Yang, K. X., Moore, P. R., Wong, C.B., Pu, J.** (2004) "An innovative solution for product service systems for consumer products", *Proceeding of Mechatronics Conference*, August 30-1 September, 2004, Ankara, Turkey.
3. **Moore, P. R., Pu, J., Wong, C. B., Chong, S. K., Yang, K. X.** (2003) "A component-based development environment for environmental life-cycle information management systems for consumer products", *International Conference on Computer, Communication and Control Technologies(CCCT)*, 2003, July 31, August 1-2, 2003, Orlando, Florida, USA.
4. **Simon, M., Allman L., Moore, P.R., Yang. K.X., Cock, T.** (2004) "Product design informed by life cycle information: initial results from ELIMA", *Proceeding of International Conference on Manufacturing Research (ICMR2004)*, September, 7-9, Sheffield Hallam University.
5. **Bodenhoefer, K., Schneider, A., Cock, T., Brooks, A., Sands, G., Allman, L., Simon, M., Chong, S.K., Yang. K.X, Delannoy, O., Catanese, B., Mueller, K.,** (2004) "Environmental life cycle information management and acquisition: first experiences and results from field trials" *Proceeding of Electronics Goes Green 2004+*, September 6-8, 2004, Berlin, Germany.
6. **Yang, K. X., Moore, P. R., Wong, C.B., Pu, J., Chong, S. K. and Simon, M** (2005) "A methodology for realising product-service systems for consumer products", submitted to the journal *IEEE Transactions on Systems, Man, and Cybernetics*.